# BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.**

2624 #3

ATTORNEY DOCKET NO.: 1952098-0003

**IN THE UNITED STATES PATENT & TRADEMARK OFFICE**

In re patent application of:

SHEKTER, Jonathan Martin



Serial No.: 09/863,025 Group Art Unit: 2624

Filed: May 23, 2001 Examiner:

Title: SYSTEM FOR MANIPULATING NOISE IN DIGITAL IMAGES

October 10, 2001

The Commissioner of Patents & Trademarks
Washington, D.C. 20231

**CLAIM OF PRIORITY**

Dear Sir:

We file herewith a certified copy of Canadian Patent Application No. **2,309,002**, which was filed on **May 23, 2000**, and on which the above United States application was based. We ask that this United States application be awarded priority rights in accordance with Section 119 of Title 35, Patents, (Public Law 593).

Respectfully submitted,

Oct. 10/2001
Date

Kevin Rillay
Agent for Applicant
Registration No. 41,559

Fasken Martineau DuMoulin LLP
Suite 4200, P.O. Box 20
Toronto Dominion Bank Tower
Toronto-Dominion Centre
Toronto, Ontario M5K 1N6
Telephone: (416) 366-8383
Facsimile: (416) 364-7813/7910

Please type a plus sign (+) inside this box →

PTO/SB/21 (08-00)
Approved for use through 10/31/2002. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE



# TRANSMITTAL FORM

*(to be used for all correspondence after initial filing)*

| | |
|---|---|
| Application | 09/863,025 |
| Filing Date | May 23, 2001 |
| First Named | SHEKTER, Jonathan Martin |
| Group Art Unit | 2624 |
| Examiner Name | |
| Total Number of Pages in This Submission | |
| Attorney Docket Number | 1952098-0003 |

**ENCLOSURES** *(check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Response
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☒ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) *(please identify below)*:

Remarks



**SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT**

| | |
|---|---|
| Firm or Individual name | PILLAY, kevin - Regn No. 41,559<br>FASKEN MARTINEAU DuMOULIN LLP, Toronto Dominion Bank Tower, Suite 4200, P.O. Box 20, Toronto-Dominion Centre, Toronto, Ontario, M5K 1N6, Canada |
| Signature | |
| Date | October 10, 2001 |

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231 on this date:

| | | | |
|---|---|---|---|
| Typed or printed name | | | |
| Signature | | Date | |

Office de la propriété intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An Agency of Industry Canada



# Bureau canadien des brevets
*Certification*

# Canadian Patent Office
*Certification*

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,309,002**, on May 23, 2000, by **JONATHAN MARTIN SHEKTER**, for "Digital Film Grain Reduction".



S. Grégoire
Agent certificateur/Certifying Officer

September 26, 2001
Date

Canada

(CIPO 68)
01-12-00

OPIC CIPO

# Abstract

Digital Film Grain Reduction

Jonathan Shekter

Reduction of film grain in digitized photographs and motion picture footage is an important problem with applications in such areas as special effects compositing, archival restoration, and fixed-bandwidth lossy compression. Current techniques widely used in practice (such as median filtering) are shown, both theoretically and practically, to be significantly suboptimal for this problem, especially for colour images. Previous work on the subject of film grain removal in particular and image noise reduction in general is extensively examined. Bayesian estimation is then proposed as a unifying model for noise reduction, and several classical techniques are rederived in a Bayesian framework. The dimensionality reduction techniques needed to make Bayesian estimation practical are discussed, the failure of certain classical colour noise reduction algorithms is explained, and the form of an ideal grain reduction technique is postulated. Finally, an algorithm based on these ideas is proposed and tested. This algorithm out-performs earlier techniques by exploiting correlations between the colour channels of an image.

# Chapter 4

# Practical Noise-Statistics Estimation

The noise statistics presented in the previous chapter were obtained from specially prepared noise sample images. In practical applications, such controlled data are usually not available. This chapter deals with the task of extracting noise parameters from a real image. While this is a difficult problem in general and many approaches have been devised for its solution, only a few simple techniques are discussed here. Luckily, these seem to be quite satisfactory for the problem at hand.

If a large enough sample of the noise affecting the image is available then any desired statistics can be computed. Sometimes it is possible to obtain such an external sample, if for example it is known at the time of image acquisition that noise reduction is to be performed. However in the general case this is not possible. Extracting a noise sample from an arbitrary input image is therefore an important subproblem, discussed in the first section of this chapter.

The second section of this chapter deals with estimation of the PSD of the extracted noise, which is required for various noise reduction algorithms. In the previous chapter the large sample sizes involved made simple averaging an appropriate technique. For the small samples available in real-world applications, more sophisticated approaches are required.

## 4.1 Noise Sample Extraction

Clearly the problem of the problem of reliably extracting noise from an image is not solvable in general, as it is of course equivalent to reliably recovering the original image. Yet, just as noise reduction is possible due to the specific features of "real" images, noise sample extraction is possible under restricted conditions. One such condition is that the sample region need not be very large, in this work perhaps 32 by 32 pixels at most. If an accurate model for the underlying image content in a region of this size can be constructed, this may simply be subtracted off from the observed noisy signal.

The simplest possible model is a constant image. Here we are in luck, for real images tend to have regions of constant or nearly constant intensity. This could be the sky or a wall or a section of skin. The mean pixel value even over such a region is an excellent estimator of the underlying constant value, even for very small windows. Thus the noise can be readily extracted, if such a region can be identified. This idea of finding constant regions is not new, dating back (at least) to Andrews [4] in 1977.

A human can of course perform segmentation of constant regions, based on image content, but an automatic method is preferable. It is not immediately clear how to algorithmically identify such regions, because while a human can recognize a "sky" or "wall" area, this is very hard for a computer to do. Fortunately constant regions have a nice property: of all possible input sub-images, constant regions exhibit the lowest expected variance. It therefore seems reasonable to look for windows of minimum variance within the noisy input image, the variance of each window being computed with respect to the mean value obtained by averaging the pixels within the window.

The desired quantity for each window to be examined is the variance with respect to the window mean, i.e.,

$$\begin{aligned} \sigma_w &= \sum_{u \in w} (x(u) - \mu_w)^2 \\ &= \sum_{u \in w} x(u)^2 - 2\mu_w \sum_{u \in W} x(u) + |w|\mu_w^2 \end{aligned} \tag{4.1}$$

Where $w$ is the window of pixels being examined. But, $\mu_w = 1/|w| \sum_{u \in w} x(u)$, so

$$\begin{aligned} \sigma_W &= \sum_{u \in w} x(u)^2 - \frac{2}{|w|} \left( \sum_{u \in w} x(u) \right)^2 + \frac{1}{|w|} \left( \sum_{u \in w} x(u) \right)^2 \\ &= \sum_{u \in w} x(u)^2 - \frac{1}{|w|} \left( \sum_{u \in w} x(u) \right)^2 . \end{aligned} \tag{4.2}$$

In the common case where the windows to be examined are rectangles centered on each pixel, the sums in the above expression can be computed together for each pixel using a box filter, a separable filtering operation which can be performed extremely quickly. This suggests a fast multi-pass algorithm for simultaneously computing the variances of windows centered on every pixel. First, a "mean image" is produced by box-filtering the input, then squared point-wise and divided by the window size. The input image is then squared point-wise and box filtered. Subtracting the squared-mean image from this produces a "variance image". Each pixel of this image indicates the variance of the pixels within a window centered at that pixel, relative to the window mean. The pixels of lowest value within the variance image indicate the positions of the windows which most likely contain image data of constant colour.

The results of this algorithm are illustrated in figure 4.1. Two test images are used, one extracted from the blue channel of a digitized film image and the other being the Lena test image with added white noise. The variance images of these are computed, shown here scaled linearly so that the maximum value appears as white. In these images, the minimum variance is never zero (black) – this is a direct result of the presence of noise. Ten sample regions of size 25 by 25 pixels are identified in each image and indicated by white squares. To obtain independent samples, these regions have been forced to be non-overlapping using the simple approach of selecting a region only if it is found not to overlap with any regions previously identified. This greedy technique is not the optimal solution to the problem of obtaining the ten lowest variance non-overlapping regions, but it appears to be adequate for this task because many regions have essentially the same variance, indicated by the large dark regions in the variance images.

In practice, one additional constraint is required. Image regions which are very bright or very dark can display clipping of pixel values due to quantization limits. For example, a very bright

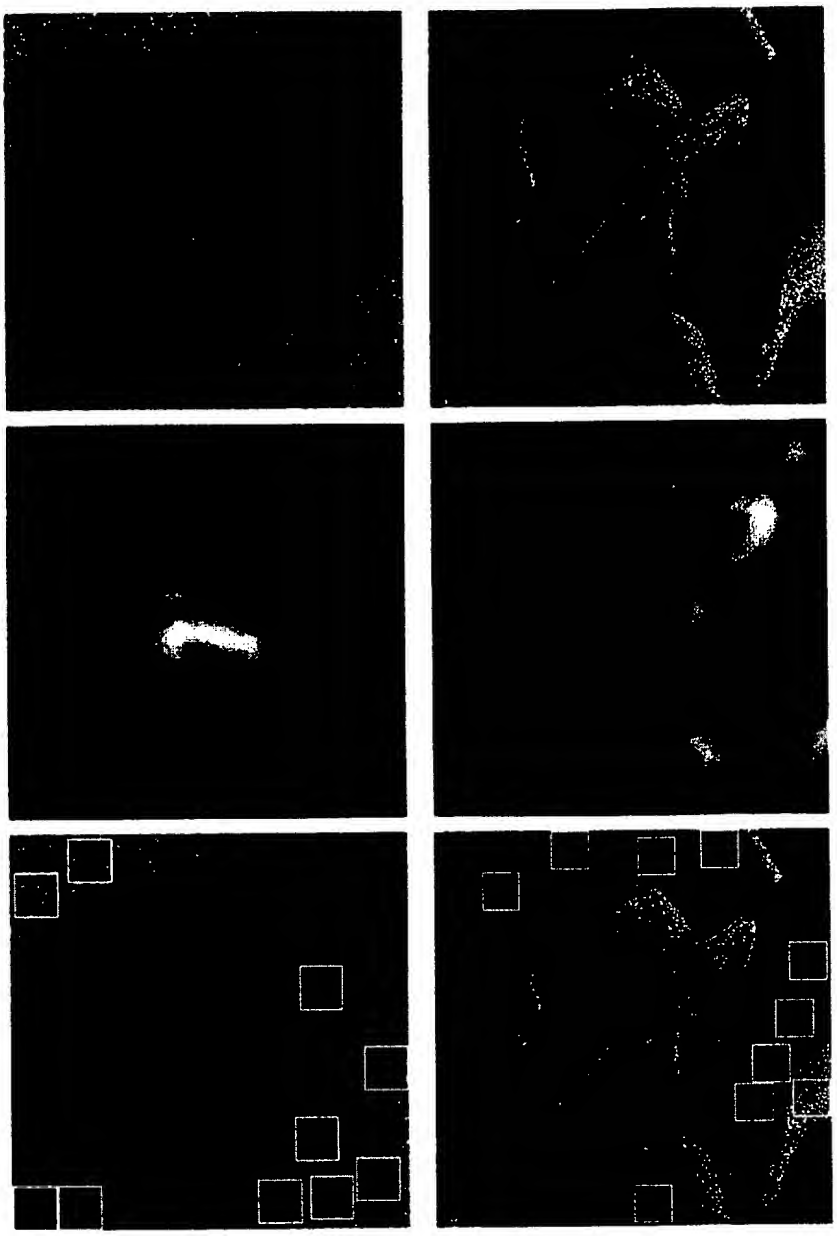


Figure 4.1: Results of variance-search algorithm to identify regions of constant color from which noise can be extracted, using a 25x25 pixel window. Top: noisy images. Middle: computed variance images. Bottom: Ten non-overlapping low variance regions selected from each image.

light source in the image can totally saturate the surrounding region of film. Depending on the details of the digitization process, this may result in a region of pixels which are all at their maximum value. Such a region will have extremely low variance, and will give a false impression of the noise level. A similar effect is possible in extremely dark regions. Therefore, care must be taken that the regions selected for noise extraction are well away from these saturation limits. Also, as seen in the previous chapter, noise levels can vary with signal amplitude. Therefore, the overall intensity of the candidate noise sample regions must be considered. The approach of requiring that the mean of each region is within one standard deviation of the overall image intensity addresses both these considerations, and seems to work well.

## 4.2 Noise PSD Estimation

Many noise reduction algorithms require an estimate of the noise PSD (or equivalently, auto-correlation function). For this reason alone spectral estimation techniques are important.

However, as discussed in the previous chapter, digital film grain appears to be adequately described by white signal dependent noise that has been degraded by the scanner point-spread function. Thus along with the noise strength $k$ (which is related to the maximum value of the PSD) and dependence exponent $D$ (which is known to be approximately $-0.5$) of equation 3.7, the noise PSD seems to be a good characterization of grain noise.

Therefore, in a theoretical sense, knowledge of the PSD can be used to determine any other desired noise statistics. Deriving analytical results relating the PSD to other statistical functions might be difficult, but the PSD can be used to synthesize an arbitrary amount of statistically accurate grain noise. Thus any statistical parameter estimation algorithm which converges given a sufficiently large noise sample can be implemented in brute force fashion.

Finally, spectral estimation is a very well studied problem, and many well-known techniques exist for the design of robust PSD estimators.

### 4.2.1 Noisy Noise Estimates

PSD estimation is not as simple as it might first seem, even for large amounts of available data. The PSD of a random signal is *not* the squared DFT magnitude of a finite portion of that signal. First, the PSD $P(f)$ is defined over all frequencies, and thus a short segment of a random process cannot contain equivalent information. Second, such a data set represents a single *realization* of the underlying random process, which can only be described statistically. This means that the DFT of a random signal is also a random signal and the squared-magnitude is an *estimator* of the true PSD, known as a "periodogram". The periodogram of a signal is not its PSD; it is instead an estimate that displays a certain amount of statistical error at each frequency.

Let $N$ be the number of input data points available, representing a finite number of samples of a random process. Surprisingly, it can be shown that the variance of the periodogram estimator (at each frequency) does not decrease as $N$ increases — a 1024 point periodogram displays just as much noise as one with only 16 points. This is because the frequency resolution of the DFT is proportional to $N$, so when more data are available the number of degrees of freedom in the periodogram output also increases. Fortunately, the mean of the estimator (at each frequency) converges to the true mean as $N$ goes to infinity. Thus the periodogram is an *unbiased* estimator, but its lack of convergence makes it *inconsistent* in the technical sense that the variance of the estimate does not go asymptotically to zero with increasing sample size.

If more data points are available, instead of using a larger periodogram estimate the data can be broken down into $K$ pieces each of length $L = N/K$ and the resulting periodograms averaged. This is an effective variance reduction technique and, for fixed $L$, the estimate converges to the true PSD as $N$ increases, because an increasing number of estimates are available for averaging. This is the "averaged periodogram" technique and was used to generate the one-dimensional spectral estimates of the previous chapter.

For those estimates, even though the PSD estimates from each of the 400 rows of the sample image were averaged, the resulting spectra still displayed appreciable variance. The situation is even worse for two-dimensional estimates. Figure 4.2 shows a 2D PSD estimate obtained by averaging together 144 independent regions of size 32x32 pixels. Even with this number of

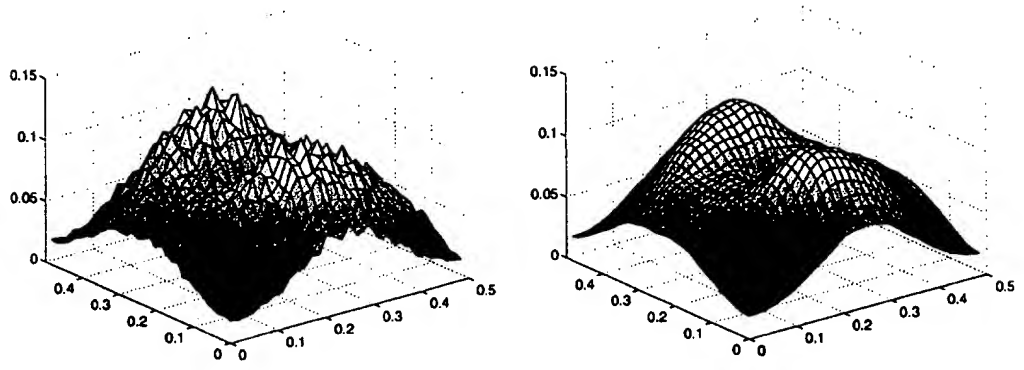



Figure 4.2: Left: Averaged periodogram estimate of the PSD of grain noise, obtained from 144 independent estimates. Right: Smoothed averaged periodogram (Blackman-Tukey estimate). The smoothing filter is Gaussian with radius of 2 pixels.

samples, the estimate is still very noisy.

Since the underlying PSD seems to be a smooth function, an intuitive solution is to try smoothing a noisy estimate with a low-pass filter. This idea actually has theoretical justification (see [32]) and is known as the Blackman-Tukey estimator. This process indeed reduces the variance of the estimate, but at the expense of bias (e.g. the value of maxima and minima will be distorted, even if the original estimate contains no noise). Even so, it is a reliable technique and often produces satisfactory results, as seen in figure 4.2. This was the technique used to generate the 2D spectral estimates in the previous chapter. Of course, in a practical situation very limited data is available. To illustrate this difficulty, figure 4.3 shows a periodogram spectral estimate derived from a single 32x32 noise patch. This is clearly a very noisy estimate, and even smoothing does not yield a satisfactory PSD.

Assuming that all sample information is being used optimally, there are essentially only two solutions to this problem. One is to increase the number of samples. This is impractical because only small noise samples can be extracted from a source image using the techniques in the previous section, which are limited by the size and number of constant color regions in the original scene. The other approach is to reduce the number of degrees of freedom in the PSD estimate. This requires modeling the PSD parametrically.

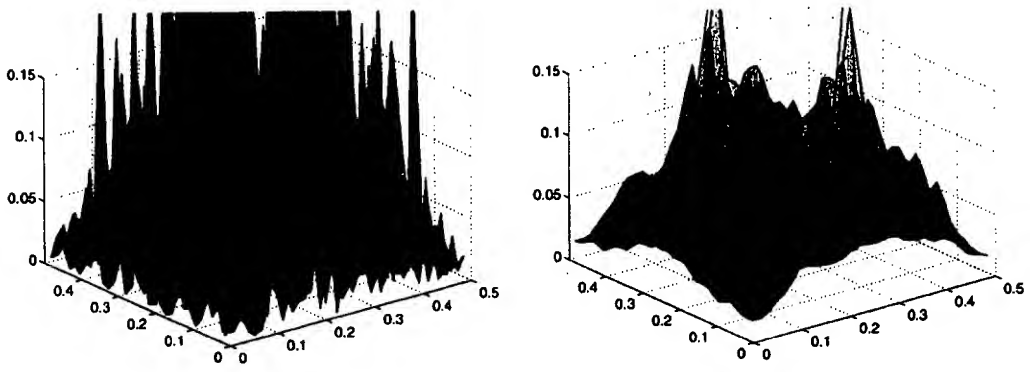



Figure 4.3: Left: Periodogram estimate obtained from a single noise sample patch. The noise is hideous. Right: smoothed single patch estimate. The noise is greatly reduced but the estimate is badly biased in places.

### 4.2.2 Parametric PSD Modeling

The trick to parametric modeling is of in course choosing the correct model and parameters. Modeling reduces the number of unknown degrees of freedom by incorporating a priori knowledge about the problem at hand, so an inappropriate model corresponds to false assumptions. Stated another way, the object of parametric modeling is the accurate description of a class of objects with as few degrees of freedom as possible.

There are a number well-developed parametric modeling approaches used in spectral estimation. The basic technique of several of these approaches is to model the random signal as filtered white noise, with the model parameters corresponding to linear filter coefficients. Although mathematically disparate, the *auto-regressive* (AR) and *moving average* (MA) techniques are closely related conceptually: the AR approach seeks to find a set of recursive filter coefficients describing an infinite impulse-response (IIR) filter, while the MA approach looks for the coefficients of convolution kernel describing a finite impulse-response (FIR) filter. Both of these techniques are characterized by an "order" which is the number of filter coefficients in the resulting model. Spectral estimation with these models is a vast topic. See [32] for details.

Unfortunately these techniques are not well developed for multi-dimensional signals. Part of this problem stems from the fact that filters involved are usually assumed to be causal.

Many one-dimensional sequences have a preferred direction (e.g. time) but there is no real equivalent to causality for 2D images. Also, many estimation techniques involve factorization of the $z$ transform of filter expressions, and most 2D polynomials cannot be factored since the Fundamental Theorem of Algebra applies only to polynomials of one variable.

However, because grain noise seems to be white noise blurred by the film scanner point-spread-function (PSF), the approach of describing the PSD by linear filter coefficients is appealing. In particular, because grain noise is only correlated very locally, the convolution kernel describing the filtering operation can only be nonzero over a small support. It is therefore reasonable to believe that this filtering operation – and hence the PSD – can be described by a small number of parameters.

Stated this way, the parametric estimation problem can be formulated as the determination of the coefficients of a convolution kernel $k(\boldsymbol{u})$ where $\boldsymbol{u}$ ranges over a small set of pixels. By the convolution theorem, the frequency response of this kernel is

$$K(\boldsymbol{f}) = \mathcal{F}(k(\boldsymbol{u})) \tag{4.3}$$

where $\mathcal{F}$ is the Discrete Fourier Transform operator. The PSD obtained by applying this filter to a unit variance white noise field is

$$P_k(\boldsymbol{f}) = |K(\boldsymbol{f})|^2 \tag{4.4}$$

where we have used the fact that the PSD of white noise is equal to its variance, in this case one. This describes the relationship between the filter coefficients and the resulting PSD model. Since the simple periodogram is known to be an unbiased estimator [1] it is reasonable to try to find the the set of filter coefficients $k(\boldsymbol{u})$ which minimizes the squared error between the model PSD $P_k$ and the observed noisy periodogram estimate $P_n$:

$$\begin{aligned} e &= (P_k(\boldsymbol{f}) - P_n(\boldsymbol{f}))^2 \\ &= (|\mathcal{F}(k(\boldsymbol{u}))|^2 - P_n(\boldsymbol{f}))^2. \end{aligned} \tag{4.5}$$

In practice the periodogram estimate $P_n$ contains a finite number of elements $N$ and can be written as a stacked vector $\boldsymbol{P_n}$. The filter kernel $k(\boldsymbol{u})$ can also be written as an $L$ by 1 vector

[1] Technically the periodogram is only asymptotically unbiased as the number of points increases, but it converges quite quickly for relatively smooth spectra. See [32] for details.

of coefficients $\boldsymbol{k}$ where $L$ is the assumed number of non-zero filter coefficients, hopefully with $L \ll N$. The DFT operation then becomes an $N$ by $L$ matrix $\boldsymbol{F}$, and the above expression can be written

$$e = (sqr(\boldsymbol{F}\boldsymbol{k}) - \boldsymbol{P_n})^T(sqr(\boldsymbol{F}\boldsymbol{k}) - \boldsymbol{P_n}) \tag{4.6}$$

where $^T$ denotes the conjugate transpose, $Tr$ is the matrix trace operator and $sqr$ is a non-linear operator which maps each element of a vector to its squared norm. Note that this expression contains quartic terms involving elements of $\boldsymbol{k}$, which will become cubic when a derivative is taken. Therefore minimization of the error is a non-linear optimization problem.

While numerical techniques certainly exist for the solution of non-linear problems, a linear formulation is desirable due to its simplicity, efficiency and numerical stability. The problem in equation 4.6 lies in squaring the $\boldsymbol{F}\boldsymbol{k}$ terms. If the squaring operation could be omitted, then this equation would represent a simple linear optimization problem, solvable by the standard least-squares solution.

By definition, the inverse Fourier transform of the PSD of a signal is the auto-correlation function. This means that if we try to model the ACF of the noise rather than the filter which generates it the squaring operation disappears and we can use linear estimation techniques. While directly modeling the PSF of the scanner which generates the observed grain noise PSD is intuitively satisfying, the ACF is just as reasonable a representation. Further, if it is assumed that the PSF is non-zero over only a few pixels then the ACF will also be non-zero over an equally small region. This effect was seen in the previous chapter where the one-dimensional ACF of grain noise was shown to be negligible beyond about three pixels.

The resulting error function is

$$e = (\boldsymbol{F}\boldsymbol{k} - \boldsymbol{P_n})^T(\boldsymbol{F}\boldsymbol{k} - \boldsymbol{P_n}). \tag{4.7}$$

It is well known that minimizing $e$ is equivalent to solving the over-determined linear system

$$\boldsymbol{F}\boldsymbol{k} = \boldsymbol{P_n}. \tag{4.8}$$

(This equation is over-determined because $\boldsymbol{F}$ is an $N$ by $L$ subset of the full DFT matrix.) The resulting estimator models the PSD as a linear combination of basis functions, which are the

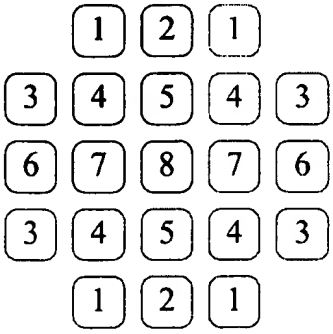



Figure 4.4: Support of the parametric ACF model developed in this section. Up to 21 pixels may be non-zero, but due to symmetry there are only eight parameters.

Fourier transforms of each possible non-zero pixel in the ACF.

The above estimator is a complex linear equation, and there is no guarantee that the resulting vector $\boldsymbol{k}$ will be real-valued. This is problematic, because the ACF is a real function. Conversely, the basis functions represented by generated by taking the DFT of pixels in the ACF need not be real, which could create a complex PSD estimate. However, the ACF is an even function. This halves the number of parameters needed to describe the ACF, but more importantly the Fourier transform of a real and even function is also real and even.

In the specific case of film grain noise, while the horizontal and vertical spectra are different as a result of a preferred direction in the film scanner, there is no reason to suppose that the scanner PSF will not display both horizontal and vertical symmetry. This symmetry is in fact observed in the PSD estimates of the preceding section, and further halves the number of parameters.

For the test data under consideration the ACF appears to extend only over about three pixels, so the support of figure 4.4 was chosen. This model allows the ACF to be non-zero over a region of 21 pixels but due to symmetry only 8 parameters are needed. For white noise only the center pixel (parameter 8) will be non-zero, but correlations up to three pixels away can be modeled with the remaining parameters, each of which corresponds to a symmetric set of non-zero pixels. The basis functions corresponding to these sets are shown in figure 4.5. Note that they have negative components, so the resulting PSD model might not be everywhere

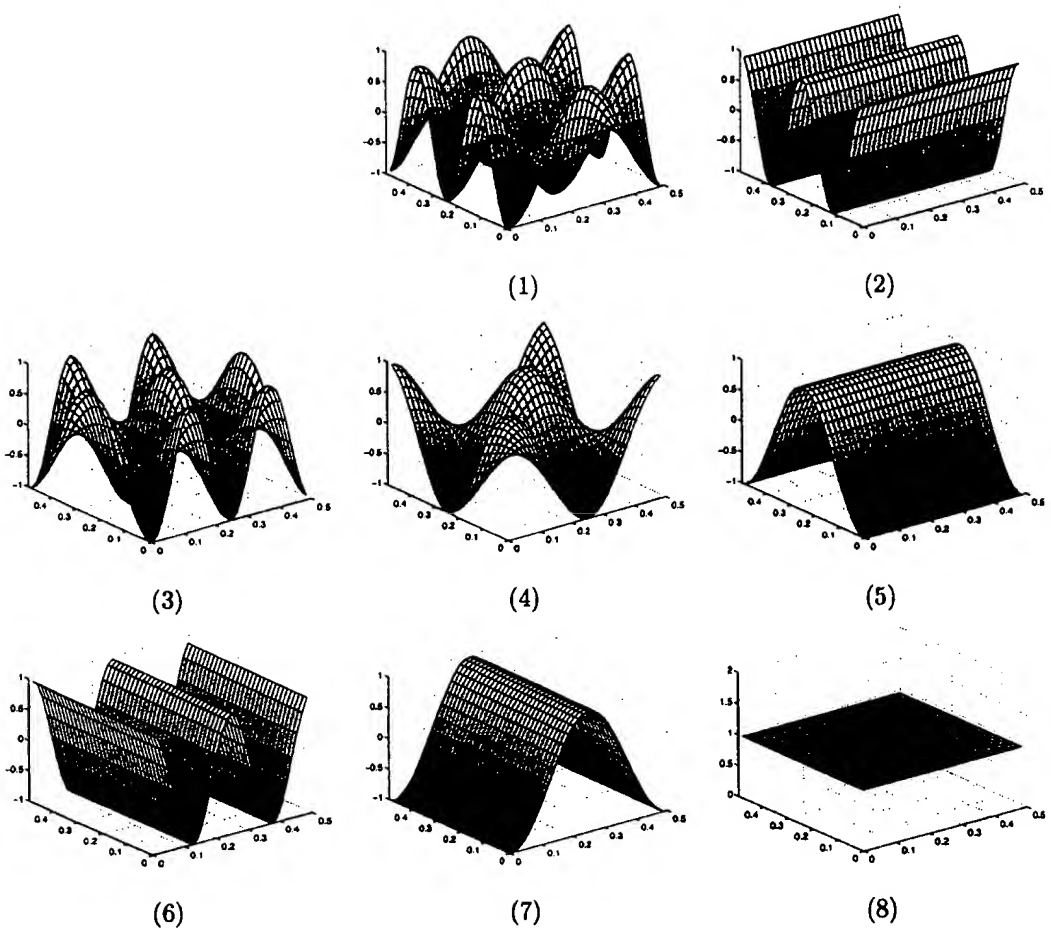



Figure 4.5: Basis functions corresponding to the eight parameters of the PSD model developed in this section. Each function is the DFT of a symmetric set of pixels belonging to the ACF.

nonnegative. Technically, the finding the optimal model parameters should be formulated constrained optimization problem to ensure that the resulting PSD is non-negative, but for the present purposes this does not seem to be a problem. In practice, negative results are rare, since we are attempting to fit an everywhere nonnegative function; and any negative values can simply be clamped to zero when the model is evaluated.

Figure 4.6 shows the result of applying this model to observed PSDs of grain noise. Fitting the model to a 144 sample averaged periodogram gives an extremely appealing model of the the grain noise PSD. More importantly, this confirms that the parametric model is expressive

enough to capture real spectra. However such a large number of noise samples will not be available in practice. Using a more realistic nine noise samples produces a PSD model which is essentially identical to that obtained from 144 samples. Even better, the quality of a model obtained from a just a single periodogram estimate is surprisingly good. This is the magic of parametric modeling: since only eight parameters need to be determined from literally thousands of input pixels, the resulting estimate is robust.

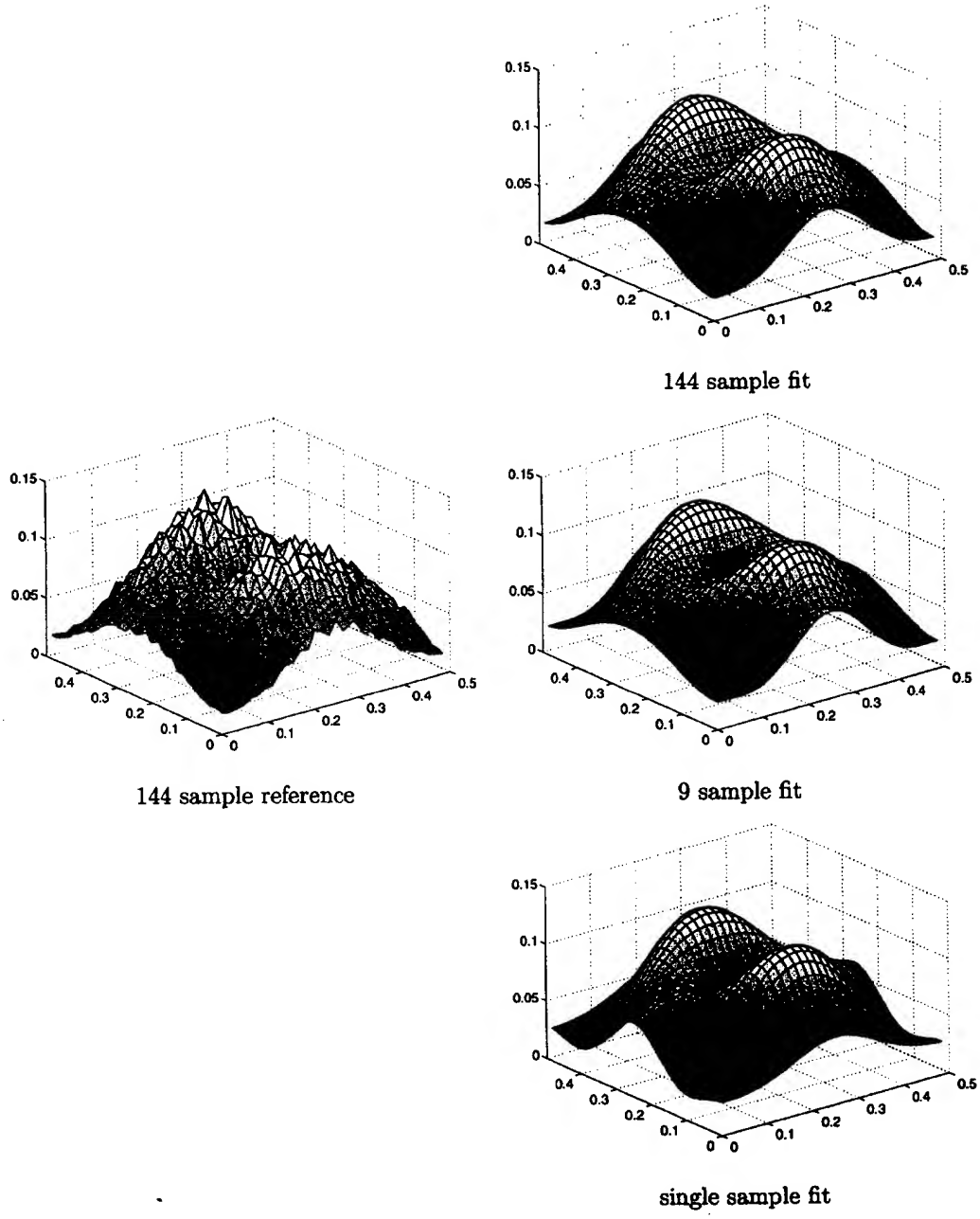



Figure 4.6: Results of applying the parametric PSD model. On the left is the reference averaged periodogram, computed from 144 32x32 noise samples. Along the right are parametric models generated by fitting to periodograms generated from various numbers of samples. Very few samples are required to produce an acceptably accurate model, in the sense of capturing the features of the reference averaged periodogram.

# Chapter 9

# Dimensionality Reduction

The Bayesian estimation approach presented previously provides an elegant theoretical solution to the problem of noise reduction. However, in principle, the joint posterior distribution required has as many variables as there are degrees of freedom in the image representation. In other words, every channel of every pixel is a variable. Therefore the first problem of designing a practical Bayesian estimator is the problem of reducing the number of dimensions which must be dealt with when working with this distribution.

It must be stated at the outset that the dimension problem is more than computational. Fundamentally, there do not exist accurate high-dimensional image models of any kind. The characterization of "real images" is a very difficult topic, and while there has been some success in modeling images from specialized domains (e.g. satellite imagery, faces, or micro-photographs of epithelial cells) no general model exists. There are of course statistical models of various kinds (Markov models, spectral characterizations, etc.) but all of these are far too broad in the sense that while they succeed in encompassing real scenes, they also include a large number of images which look like nothing recognizable. Research on image models is ongoing, but at the present time there do not exist any models which are even close to comprehensive enough for our purposes.

Further, even if such models existed, the evaluation of an integral over literally tens of thousands of dimensions poses a severe computational problem. Numerical integration algorithms scale exponentially in the number of dimensions, and analytic evaluation of a function which

takes thousands of parameters is not likely to be much better.

In contrast, low-order integrals are relatively easy to evaluate, and the problem of generating accurate analytic or numerical models of the distribution of one or a few variables has been well studied and is more or less solvable [57, 51]. This is because functions of low-dimension contain much less information and generally have much less complex distributions than their high-dimensional counterparts. For example, a single pixel of a colour image conveys almost no information about the image from which it is extracted to a human observer, which suggests that there may exist simple approximations to the corresponding three-dimensional distribution.

Hence, for lack of satisfactory models, and because of computational constraints even if such models existed, the number of variables estimated simultaneously must be very small, perhaps three at most. This problem is so severe that virtually all the techniques employed in practical noise reduction are dimensionality reducing tactics of one sort or another. For this reason, one of the aims of this chapter is to attempt to provide a unifying framework in which to study the effect and efficacy of dimension reducing techniques and associated low-dimensional estimators.

## 9.1 Window Size

The first and perhaps most obvious dimension reduction technique is to reduce the number of pixels which are considered simultaneously by examining only a small region of the input image at any one time. This is the technique of windowing, first discussed in chapter 5.

Windowing works because it is not necessary to examine the entire image before a single pixel can be successfully restored. Conversely, it is clear that no noise reduction algorithm can operate without examining nearby pixels. This raises the question of how many surrounding pixels are really necessary for good performance. Put another way, how big does the window have to be? Windows which are too large are computationally infeasible, while windows which are too small impair noise reduction performance.

Figure 9.1 shows a sequence of images centered around a single noisy pixel. This pixel is in fact part of an eyelash, but it is not possible to discern this fact until the window gets to be about 25 by 25 pixels. After this point, further increases in window size do not really seem to

contribute anything to knowledge of the pixel in question.

From a more formal point of view, the range over which the image auto-correlation function has appreciable magnitude is of interest. Technically, images are not stationary so the auto-correlation function is not really well defined, nor does zero correlation imply statistical independence. Nonetheless, estimates of this function provide a useful practical measure of the range over which pixels are closely related. Many authors have observed that the ACF of real images tends not to extend past 20-30 pixels at most; scanned film images seem to be no exception. Figure 9.1 shows a one dimensional slice of ACF of the eye-test image taken in the horizontal direction. This function has appreciable magnitude until at least five pixels, and is negligible at about 15-20 pixels. Remembering that the ACF extends in both directions, this explains why an 11 pixel window was needed to resolve the eyelash feature, while windows larger than about 25 pixels seemed to be unnecessary.

As further proof, figure 9.1 demonstrates that, at least the case of film grain removal, a reasonably large window is absolutely necessary. In this figure, two 11 by 11 pixel windows are compared. One of these is again a portion of the eyelash. The other is from a section of smooth skin which contains no image detail, but shows a distinctive noise pattern. With this amount of context, it is not possible to tell that one of these windows represents an image feature while the other is just noise. This demonstration makes it quite clear that fairly large windows are required for film grain reduction. Without such context, either image features will not be correctly identified (resulting in blurring) or noise patterns will be mistaken for detail (resulting in artifacts). It is partially for this reason that the median-type filters – which operate only over very small windows – were dismissed so quickly in previous chapters.

Conversely, the observation that there does not seem to be much point in using windows larger than a certain size (perhaps 25 pixels square for the eye test image) is of tremendous value. It prevents the dimension of the PDF from scaling as the number of pixels in the image, which makes linear-time restoration algorithms theoretically possible, for example.

It is also a good justification for employing a block-by-block algorithm. Strictly speaking, the quality of the estimation of pixels near the edges of a block must suffer, since correlations with nearby pixels outside of the block are not taken into account. However, if a sufficiently

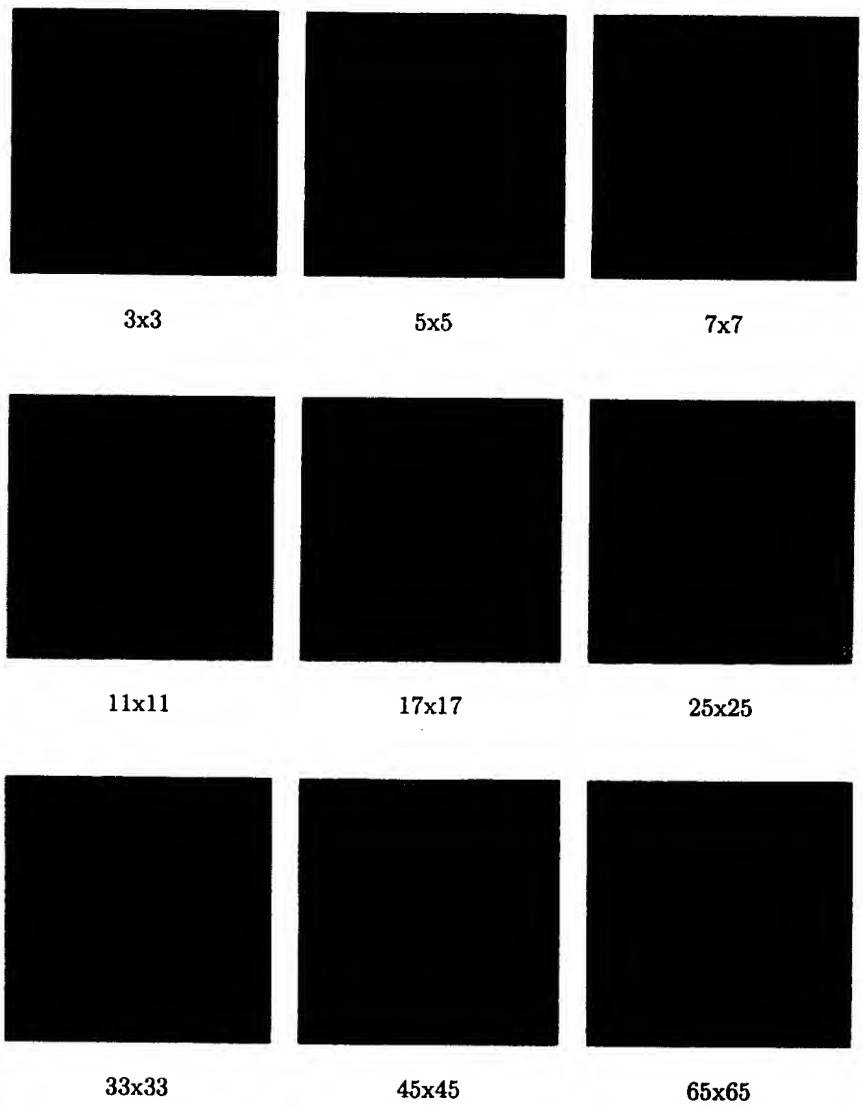



Figure 9.1: Varying window sizes centered around a single pixel. Although this pixel is part of an image feature, it is not possible to discern this until the window gets big enough, after which further context does not help much.

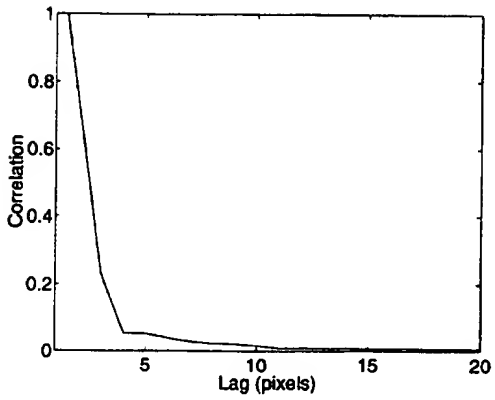



Figure 9.2: A horizontal cross-section of the auto-correlation function of the eye test image.

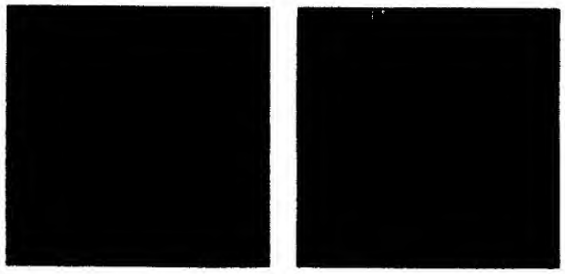


Figure 9.3: Two 11 by 11 pixel windows. Which contains an image feature and which is just noise?

large central region of each block contains an acceptable estimation, then the blocks can be overlapped in a fashion which provides roughly uniform estimation error at each pixel.

Of course, the problems with block processing can be avoided entirely through the use of transforms based on localized kernels, e.g. subband transforms. The use of such transforms is again justified by that fact that only a finite window needs to be examined to compute any one pixel, and it is precisely this locality property which makes subband transforms computable in time proportional to the number of pixels (linear time). Algorithms based on such transforms essentially operate on a pixel by pixel basis, yet take into account a region of pixels as large as the support of the (effective) kernels used to generate the transform.

## 9.2 Separable Approximations

### 9.2.1 The Effect of Independent Estimation

Consider the standard minimum-variance Bayesian estimator of equation 8.13, repeated here

$$\tilde{\boldsymbol{x}} = \int \boldsymbol{x}\, p(\boldsymbol{x}|\boldsymbol{y})\, d\boldsymbol{x}. \tag{9.1}$$

Strictly speaking, the dimension of the required posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ cannot be reduced. However, if the vector $\boldsymbol{x}$ can be written as $\boldsymbol{x} = \left[\begin{smallmatrix}\boldsymbol{x_1}\\ \boldsymbol{x_2}\end{smallmatrix}\right]$ such that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are statistically independent, then $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ may be estimated independently of one another. Mathematically, in this case the PDF can be factored into a product of functions of lower dimension, a property known as *separability*:

$$p(\boldsymbol{x}|\boldsymbol{y}) = p_1(\boldsymbol{x_1}|\boldsymbol{y})\, p_2(\boldsymbol{x_2}|\boldsymbol{y}). \tag{9.2}$$

The Bayesian estimation integrand can then be written as a product of two functions over different variables, allowing independent evaluation:

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \int \boldsymbol{x}\, p_1(\boldsymbol{x}_1|\boldsymbol{y})\, p_2(\boldsymbol{x}_2|\boldsymbol{y})\, d\boldsymbol{x} \\ &= \begin{bmatrix} \left(\int \boldsymbol{x}_1\, p_1(\boldsymbol{x}_1|\boldsymbol{y})\, d\boldsymbol{x}_1\right)\left(\int p_2(\boldsymbol{x}_2|\boldsymbol{y})\, d\boldsymbol{x}_2\right) \\ \left(\int p_1(\boldsymbol{x}_1|\boldsymbol{y})\, d\boldsymbol{x}_1\right)\left(\int \boldsymbol{x}_2\, p_2(\boldsymbol{x}_2|\boldsymbol{y})\, d\boldsymbol{x}_2\right) \end{bmatrix} \\ &= \begin{bmatrix} \int \boldsymbol{x}_1\, p_1(\boldsymbol{x}_1|\boldsymbol{y})\, d\boldsymbol{x}_1 \\ \int \boldsymbol{x}_2\, p_2(\boldsymbol{x}_2|\boldsymbol{y})\, d\boldsymbol{x}_2 \end{bmatrix}\end{aligned} \tag{9.3}$$

where the fact that all probability distributions integrate to one has been used.

Unfortunately, complete independence of variables is rare, so in most cases all unknown quantities should really be estimated simultaneously. However, this is not even remotely possible, given that the dimension of the estimation integral must be very low. In practice we can only estimate a handful of variables at once. Essentially, we must perforce ignore most dependencies between variables.

Given this harsh reality, we are obliged to examine the effect of independent estimation of variables which are not statistically independent. For simplicity, consider the problem of estimating just two scalar variables $x_1$ and $x_2$. The optimal estimator is of course

$$\begin{aligned} x_1 &= \int x_1\, p(x_1, x_2)\, dx_1 dx_2 \\ x_2 &= \int x_2\, p(x_1, x_2)\, dx_1 dx_2 \end{aligned} \tag{9.4}$$

(for notational simplicity, it is assumed that $p(x_1, x_2)$ is conditioned on the value of the observed variables, i.e. $p(x_1, x_2) = p(x_1, x_2|\boldsymbol{y})$.) If these variables must be estimated independently, it is because using the full posterior distribution $p(x_1, x_2)$ is impractical. Instead, the lower dimensional marginal distributions $p_{x_1}(x_1)$ and $p_{x_2}(x_2)$ must be used:

$$\begin{aligned} x_1 &= \int x_1\, p_{x_1}(x_1)\, dx_1 dx_2 \\ x_2 &= \int x_2\, p_{x_2}(x_2)\, dx_1 dx_2. \end{aligned} \tag{9.5}$$

Reversing the derivation of equation 9.3, this is equivalent applying the optimum estimator in the case where

$$p(x_1, x_2) = p_{x_1}(x_1) p_{x_2}(x_2). \tag{9.6}$$

Thus we get the important result that independent estimation of correlated variables is equivalent to replacing the joint posterior distribution by the product of the marginal distributions.

There is no reason to believe that the product of the marginal distributions is a reasonable approximation to the joint PDF. In general it is actually quite a poor substitute. For example, this is why it is not possible to perform noise reduction by acting only on single pixels taken individually. But, there is also no reason that the input variables must be the ones that are estimated. Rather, estimation can be performed in some transformed domain. Nor is it necessary to use the true marginal distributions $p_{x_1}(x) = \int p(x_1, x_2)\, dx_2$ and $p_{x_2}(x_2) = \int p(x_1, x_1)\, dx_1$. Instead, any other univariate distributions may be used for the independent estimation of $x_1$ and $x_2$, if doing so produces a better approximation to the joint PDF. In other words, we are forced to use a separable approximation, but we are able construct this approximation arbitrarily.

Thus dimension reduction can be viewed as an exercise in the construction of separable approximations to the full joint posterior PDF.

### 9.2.2 The Meaning Of Correlation

Before decorrelating transforms can be discussed, the properties of correlation must be properly appreciated. While there seems to be some confusion in the literature about the significance of this quantity, the definition is simple enough: correlation is defined between two random variables $x$ and $y$ as the expected value of their product $\mathrm{E}\{xy\}$. Correlation is indeed a very useful quantity, but the its meaning is more subtle than is usually realized.

It is easy to show that if two zero-mean random variables $x$ and $y$ are independent then they are uncorrelated. Taking the expectation of their product yields the integral

$$\mathrm{E}\{xy\} = \int\int xy\, p(x, y)\, dx\, dy. \tag{9.7}$$

Using the fact that statistical independence implies separability of the joint PDF we have

$$\begin{aligned} \mathrm{E}\{xy\} &= \int\int xy\, p_x(x) p_y(y)\, dx\, dy \\ &= \left(\int x\, p_x(x)\, dx\right)\left(\int y\, p_y(y)\, dy\right) \\ &= \mathrm{E}\{x\}\,\mathrm{E}\{y\} \\ &= 0 \end{aligned} \tag{9.8}$$

where the zero-meanness of $x$ and $y$ has been used.

This result also implies that two correlated (zero-mean) variables cannot be statistically independent. Thus we can speak of an algorithm exploiting the "correlation" between nearby pixels, by which is meant that such an algorithm takes advantage of statistical dependence between input variables. In this positive sense the common loose usage of the term to mean "statistical dependence" is correct.

However, the above derivation does not go through in reverse. That is to say, just because two variables have zero correlation does not mean that they are statistically independent. *"Uncorrelated" does not mean "statistically independent."* Two variables are independent only if their joint probability density is separable, whereas there are a great many two-variable functions such that the integral of equation 9.7 is zero. For example all radially symmetric functions have this property. More generally the symmetry condition $p(x, y) = p(-x, -y)$ is sufficient (but not necessary) to guarantee that 9.7 evaluates to zero, and there are certainly many non-separable functions with this property.

Note that the multi-variate Gaussian distribution is separable, because $\exp(x^2 + y^2) = \exp(x^2)\exp(y^2)$. Thus, if two jointly Gaussian random variables are uncorrelated, they are independent. Techniques which assume that decorrelating two variables produces statistical independence are not incorrect if the variables in question are Gaussian.

### 9.2.3 Constructing Separable Approximations

As hinted earlier, an arbitrary joint PDF $p(x, y)$ may be separable in variables other than $x$ and $y$. For example all radially symmetric functions are separable in polar coordinates. More generally, it may be possible to to find (continuous, one to one) transformations $S(x, y) \mapsto u$

and $T(x,y) \mapsto v$ such that

$$p(x,y) = p_u(S(x,y))p_v(T(x,y)) \tag{9.9}$$

for some PDFs $p_u$ and $p_v$.

One simple case is when $S$ and $T$ are both linear transformations. In this case the underlying PDF is separable along some set of axes which are not aligned with the observed variables $x$ and $y$. It is therefore possible to apply an inverse transformation which in effect rotates the PDF to produce uncorrelated variables $u$ and $v$. Decorrelation via linear transformation is an appropriate technique if it is expected that certain variables are independent in linearly transformed coordinates. Really, this just means that the joint PDF in question is separable along some set of axes.

Unfortunately, this is still quite a restrictive condition. The multi-variate Gaussian distribution is the only standard PDF which falls into this category. It will more usually be the case that the joint PDF is not separable along any axes. It might still be possible to find a (non-linear) transformation which produces independent variables, but there is no general technique for this (although the case of radial symmetry should be considered.)

However it may be the case that the joint probability density function is "approximately" separable. Once again, there is no reason to assume that the joint PDF is naturally aligned to the coordinate axes defined by $x$ and $y$. Instead, we are free to construct a separable approximation in any transformed space, again using the transformation $u = S(x,y)$ and $v = T(x,y)$. The marginal densities of the transformed variables are then $p_u(u)$ and $p_v(v)$ and the resulting separable approximation is $p_{uv}(u,v) = p_u(u)p_v(v)$. Thus $p(x,y)$ is replaced by $p_u(S(x,y))p_v(T(x,y))$, which may be a better approximation than $p_x(x)p_y(y)$. Note that this approximation is never explicitly constructed, nor must the full form of the joint PDF be known at all. Rather, any algorithm which estimates transformed variables independently implicitly assumes that the PDF is of such a form, and acts as if this were the case.

It is also possible, as noted earlier, to choose arbitrary distributions in place of the true marginal densities, which may yield a better approximation. However the choice of such functions is a non-linear optimization problem of extremely high dimension, and it is not clear if the

resulting approximations can be appreciably better than those constructed from the marginal distributions. More to the point, due to the lack of accurate image models, the full PDF which one would wish to approximate is not available. For these reasons, this intriguing possibility is left as an open question in this work, and approximation using marginal densities is assumed throughout.

Under these constraints, the problem becomes one of choosing the optimal set of axes along which to construct a separable approximation. Such an approximation involves integrating along each axis to generate the required marginal densities, so this problem can be thought of as the finding the directions such that the "shadows" of the joint PDF along these directions yield the most information. (Actually, this problem is well studied from the point of view of finding the projection of a high dimensional PDF which best reveals the form of the distribution, a good survey of which is found in Scott [51]. These techniques are not suited to the current problem, since rather than interpreting the PDF we wish to approximate it.)

Consider a non-separable joint PDF which is elongated in a particular direction not aligned with either axis. Such a PDF is shown in figure 9.2.3 as (a) a 3D surface and (b) a contour plot. This PDF is a function of the distance to a line segment of length 1/2 which has been rotated by 30°, and is not a separable distribution under any coordinate transformation. The marginal densities along each axis are also plotted; due to normalization they extend higher than the joint PDF. Parts (c) and (d) of the same figure show the resulting separable approximation obtained by multiplying the marginal distributions. Note that the orientation of the original PDF is not at all captured. The sharp peak is lost, and the approximation is axis-aligned, as all separable approximations must be. By comparison, figure 9.2.3 (a) and (b) show the same PDF in a coordinate system aligned with the long axis of the distribution. The resulting separable approximation, shown in parts (c) and (d), while not perfect, is a reasonable match to the original function. In particular it has the right extents in each dimension, and the sharp peak is much better preserved.

The importance of the correct selection of coordinate system in constructing separable approximations is now clear. Again, in practice this approximation is never explicitly constructed, but an algorithm which employs only marginal distributions actually operates on the joint PDF

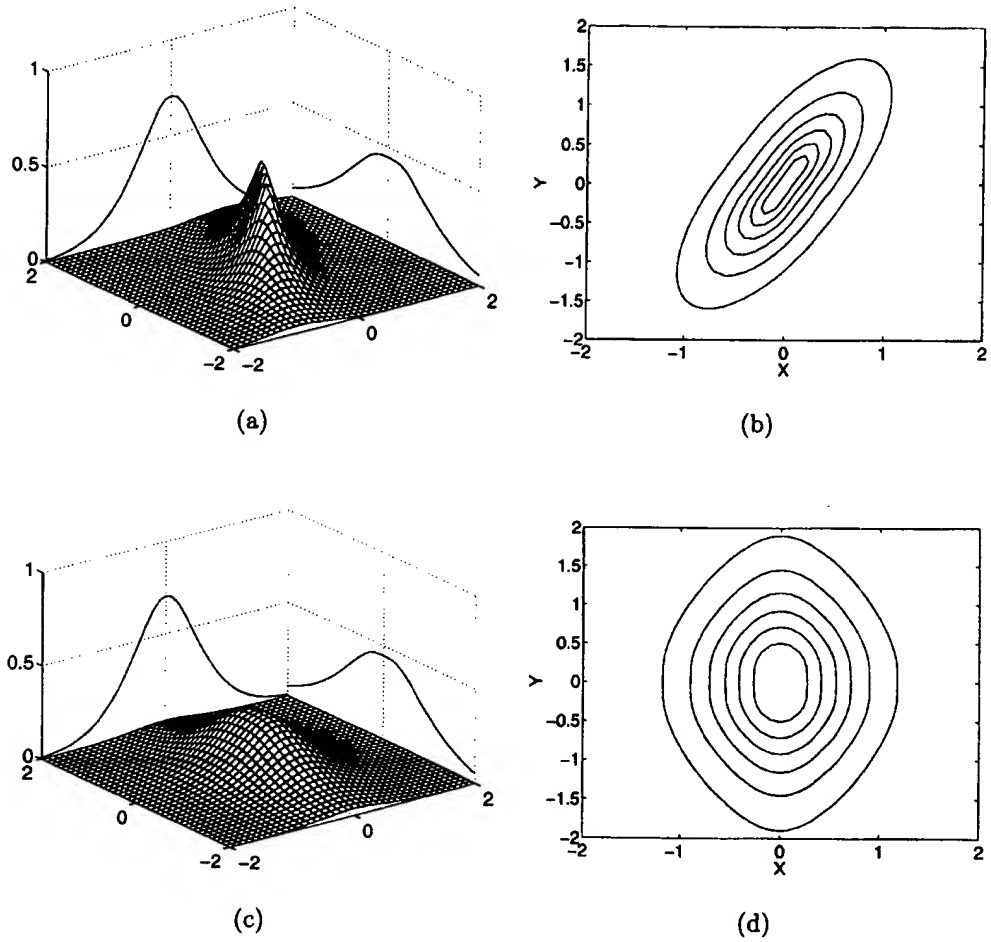



Figure 9.4: Construction of a separable approximation to a joint PDF with poor choice of axes. The resulting model fails to capture important features of the distribution.

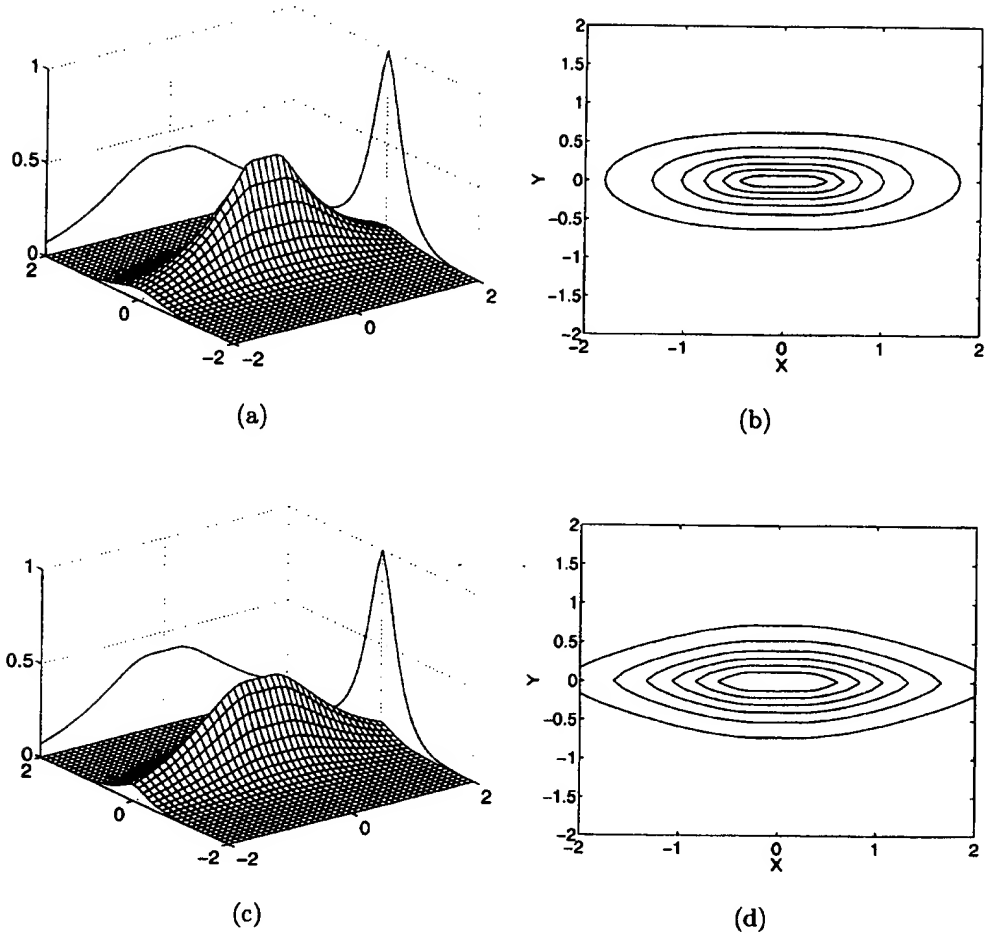



Figure 9.5: Construction of a separable approximation to a joint PDF along axes which decorrelate the two variables. The resulting model is a good approximation and captures the major features of the original distribution.

described by their product. Hence if independent processing of variables must be performed, it is essential that the correct variables are chosen.

Although general analytical results on the optimal set of axes for separable approximation are difficult to obtain (and we have been unable to find any such results in the literature) it turns out that the orthogonal coordinate system in which $u$ and $v$ are uncorrelated is usually a good choice. To see why, consider that any elongated PDF which is symmetric about its major axis will display zero correlation between variables only when rotated to align exactly with the coordinate axes. In general the transform which decorrelates two variables aligns the "major axis" with a transformed variable, thus it is a reasonable coordinate system for the construction of a separable approximation.

This fact finally explains the importance of decorrelating transformations. While it is not usually possible make two variables independent through a linear transformation, it is always possible to apply a rotation which removes their correlation. This is a consequence of the fact that zero correlation does not imply statistical independence. Such a transformation is called the Karhunen-Loeve Transform (KLT), which will be discussed in detail in section 9.3.2.

Finally, all of this discussion and all results generalize immediately to higher dimensions. In that case, the Karhunen-Loeve Transform simultaneously removes the correlation between all pairs of variables. The "major axes" of the (high-dimensional) distribution will end up aligned with transformed variables, facilitating the construction of separable approximations. Actually, the KLT can be shown to choose the orthogonal axes along which the distribution displays maximum variance [37] so the phrase "major axes" has a precise and satisfying interpretation. And of course, if the PDF really *is* separable along one more axes in some coordinate system, such a transform will produce as many statistically independent variables as possible.

In short, in the case where a set of variables must be processed independently it is useful to first employ a linear transformation to decorrelate the variables, even when their joint PDF is not separable, because the transformed variables will be "more" separable and thus "closer" to statistical independence.[1]

[1]It is obvious that a great deal more work, both analytical and experimental, is needed on this topic before such a statement can be made precise; oddly enough the author has been unable to find any such studies. This may be because the signal processing community is only just beginning to fully appreciate that second-order

## 9.3 Image Transforms

An image transform in this context refers to a linear transformation applied to the input values comprising the image. In the multi-channel case it may in general involve terms which operate between image channels.

As discussed previously, there is no reason to believe that raw pixel values are the best domain for the application of image processing techniques. The problem is again one of dimensionality: any Bayesian estimation technique must eventually evaluate the minimum-variance integral of equation 8.13, and for practical reasons this integral must be taken over a space of very low dimension. This corresponds to the simultaneous estimation of at most a few different variables. With this constraint, the choice of variables becomes critical. In fact, independent processing of pixel values yields very poor estimators, so image transform techniques are more or less required in practice.

There are essentially three different ways in which image transforms have been exploited for noise reduction in the past; these are in some sense different approaches to the same problem, justified in different ways.

1. Decorrelation of input values. With a firm understanding of what correlation is and is not, we can proceed to examine the rich field of image transformations designed to decorrelate the input data. As discussed above, decorrelation is valuable for noise reduction because it makes the input variables (hopefully) "close" to statistically independent. This justifies the independent estimation of only one or a few simultaneous values.

2. Energy compaction. Certain types of (energy-preserving) transforms tend to concentrate the signal energy in just a few large coefficients. Essentially, energy compaction acts to remove redundancy between elements of the input data, which is a sort of "soft" dimension reduction. Since noise cannot be so compacted, energy compacting transforms can be used for noise reduction by the technique of coring, discussed in section 5.5.2. This property can also be exploited for image compression (see e.g. Jain [29])

(Gaussian) models are not good descriptions of many phenomena.

3. Feature Detection. If only a few variables can be processed simultaneously, it is important that each variables convey meaningful information in some sense. Rather than trying to estimate single pixel values, a higher level approach is warranted. The transformed variables might therefore represent image *features* such as edges or regions of certain shape and colour. This changes the estimation problem from "what is the value of this pixel?" to "how visible is this feature?" In effect, feature detection expands the number of pixels corresponding to each variable, providing a sort of "context" which may be very helpful if each variable must be estimated independently.

These three approaches are inter-related. For example, energy compaction can be achieved if the transform employed is able to represent the image by a few crucial "features", and it can be shown that the orthogonal transform which provides maximum energy compaction is the one which completely decorrelates the input variables (the KLT). Many transforms simultaneously achieve most or all of these objectives; the difference is more in how the resulting coefficients are interpreted than in the transformations themselves.

A unifying interpretation again comes from considering independent processing as the construction of a separable approximation to the joint PDF. Viewed this way, all transform techniques seek to find a transformed set of variables $z$ such that

$$p(z) \approx p_{z_1}(z_1)p_{z_2}(z_2)\dots p_{z_n}(z_n) \tag{9.10}$$

where the $z_i$ are disjoint sets of variables which are each processed independently. The differences between the various transform types can be interpreted in terms of how this approximation is constructed. Decorrelation is a general approach which aligns variables with the "major axes" of the joint PDF, energy compaction techniques seek to find a transform which minimizes the complexity of the resulting approximation, and feature-based transforms employ *a-priori* knowledge of the important aspects of the joint PDF.

### 9.3.1 Fourier Techniques

The material in this section has actually been encountered before, in the derivation of the single and multi-channel Wiener filters of sections 5.2.2 and 6.2. There it was explained that

the Discrete Fourier Transform is used to diagonalize the dense covariance matrix to allow practical computation of the discrete Wiener filter. An alternate view is that this diagonalization removes the correlations between image pixels, justifying independent processing of each transform coefficient (frequency).

The first question to be answered is, what does it mean for two "pixels" to be correlated? Correlation is defined as the expected value of the product of two random variables, which implies the existence of an ensemble of possible values for each variable. In other words, the correlation between the pixels of an image can only be defined in terms of a probabilistic image model. It is not possible to "remove the correlation" from a single image; it is only possible to decorrelate the pixels of an *ensemble* of images defined by some high dimensional multi-variate PDF, i.e. an image model.

If the image model is assumed to be stationary, then the correlation between any two pixels depends only on the distance between these pixels. Despite the fact that images are not really stationary (!) this assumption is invariably made in practice. It is one thing to note that the "statistics" of an image vary with position, but quite another to attempt construction of a model which accounts for this. As noted above, "statistics" only apply to an ensemble of images, so what is the ensemble which has the same local variations as the particular image in question? In practice this question is usually unanswerable.

We are therefore more or less forced to assume that the observed image displays stationary statistics, at least as far as computing correlation coefficients is concerned. Of course, as discussed in section 5.3, one can always break the image in windows or segments which are individually assumed to be stationary, so the assumption of stationarity is not as dire as might initially be thought.

Once the assumption of stationarity is made, the auto-correlation function (ACF) uniquely defines the correlation between any two pixels as a function of distance. For colour images, there will be one ACF for each channel plus a cross-correlation function (CCF) for each unique pair of channels which describes the correlation between these channels as a function of spatial separation. If the ACF/CCF are known, a covariance matrix (between all pairs of image pixels and channels) may be produced. In the single channel case this will be a (block) Toeplitz matrix

due to the stationarity of the image statistics, as discussed in section 5.2.3. In the multi-channel case, the resulting matrix will have a block structure defined by the image channels, where each block is Toeplitz.

The resulting covariance matrix is far too large to be computed explicitly for most problems. However the ACF/CCF provides equivalent information, and this can be estimated using a wide variety of techniques (see e.g. chapter 4 or Kay [32]). But, in the single channel case, if we are willing to assume that the input image displays stationary statistics and is periodic, there is no need even to do this much explicitly: under these assumptions the resulting covariance matrix is circulant, and all circulant matrices are diagonalized by the DFT. For single channel images which are assumed stationary and periodic, the DFT diagonalizes the image covariance matrix. For multi-channel images, if each channel is assumed to be stationary and periodic, then each block of the covariance matrix which describes the (cross-)correlation between two channels is circulant and thus diagonalized by the DFT.

This analysis explains the popularity of Fourier-based techniques in image processing: under very general assumptions the DFT decorrelates the pixels of a single image channel. However DFT-based techniques suffer from a number of problems in practice. Fundamentally these stem from the fact that diagonalization of the covariance matrix does not truly produce independent variables, if the input does not have a multi-variate Gaussian distribution. In the case of the DFT, there are strong correlations between frequency components if the image contains any feature which spans many frequencies, such as sharp edges. Independent processing of each frequency cannot detect such features, a shortcoming which may result in blurring or ringing artifacts.

### 9.3.2 Karhunen-Loeve Transform and Approximations

The Karhunen-Loeve transformation has been encountered before in this work. It is the unique transformation which completely removes all correlations between a set of input variables. As discussed in section 9.2.2 this has important implications for algorithms which must operate on transformed variables independently.

Once again let $\boldsymbol{R}_{\boldsymbol{x}}$ be the covariance matrix between all elements of the input vector $\boldsymbol{x}$. We

seek a linear transformation of $\boldsymbol{A}$ which removes all correlations between the elements of $\boldsymbol{x}$, i.e. a matrix $\boldsymbol{A}$ such that

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{9.11}$$

and

$$\mathrm{E}\{y_i y_j\} = \begin{cases} \sigma y_i^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9.12}$$

It is clear from this definition that the covariance matrix $\boldsymbol{R_y}$ of the transformed vector $\boldsymbol{y}$ is diagonal, that is

$$\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^T\} = \boldsymbol{R_y} = \boldsymbol{\Sigma} \tag{9.13}$$

where $\Sigma$ is a diagonal matrix containing the variances $\sigma y_i^2$ of the transformed elements of $\boldsymbol{y}$. Substituting in equation 9.11 this becomes

$$\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^T\} = \mathrm{E}\{(\boldsymbol{A}\boldsymbol{y})(\boldsymbol{A}\boldsymbol{y})^T\} = \boldsymbol{\Sigma} \tag{9.14}$$

Using the linearity of expectation and the fact that $\boldsymbol{R_x} = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\}$ we finally obtain the defining expression

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{R_x}\boldsymbol{A}^T \tag{9.15}$$

Which states that $\boldsymbol{A}$ diagonalizes the covariance matrix $\boldsymbol{R_x}$. Since a covariance matrix is symmetric the Principal Axes Theorem applies and this equation is solvable by standard eigenvector techniques. In fact, because the resulting $\boldsymbol{A}$ is orthonormal we have $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$ and equation 9.15 may be rewritten as

$$\boldsymbol{R_x}\boldsymbol{A}^T = \boldsymbol{A}^T\boldsymbol{\Sigma} \tag{9.16}$$

which shows that the columns of $\boldsymbol{A}^T$ are the eigenvectors of $\boldsymbol{R_x}$ while the diagonal matrix $\boldsymbol{\Sigma}$ contains the eigenvalues. Since this matrix also describes the variances of the elements of $\boldsymbol{y}$, these variances are in fact the eigenvalues of $\boldsymbol{R_x}$.

The resulting matrix $\boldsymbol{A}$ defines the Karhunen-Loeve transformation with respect to the covariance matrix $\boldsymbol{R_x}$. This is an important point: unlike the DFT or other transforms there

is no single "Karhunen-Loeve Transformation" described by a fixed matrix . Rather, the KLT is defined as that transformation which diagonalizes the covariance matrix $\boldsymbol{R_x}$; the exact value of this transformation is therefore strongly dependent upon $\boldsymbol{R_x}$.

Aside from perfect decorrelation, the KLT is optimal in the sense of energy compaction. Let the transformed variables $y_1, y_2, \ldots y_N$ be ordered by decreasing variance, i.e.

$$\sigma y_1^2 \geq \sigma y_2^2 \geq \ldots \geq \sigma y_N^2 \tag{9.17}$$

where $\sigma y_i^2 = \mathrm{E}\{y_i^2\}$. This re-ordering of the variables can in fact be "built-in" to the KLT transformation matrix by simple permutation; it is just reordering of the eigenvalues of $\boldsymbol{R_x}$. With this modification the KLT is also known as the Principal Components transform. Consider now any other orthonormal linear transformation $\boldsymbol{z} = \boldsymbol{B}\boldsymbol{x}$ applied to the data. This produces variables $z_1, z_2, \ldots z_N$ which can of course be ordered by their variances

$$\sigma z_1^2 \geq \sigma z_2^2 \geq \ldots \geq \sigma z_N^2 \tag{9.18}$$

Then, the KLT has optimal energy compaction in the sense that for any $n \leq N$

$$\sum_{i=1}^{n} \sigma y_i^2 \geq \sum_{i=1}^{n} \sigma z_i^2 \tag{9.19}$$

This inequality can be interpreted as a sort of inductive statement of optimality. Setting $n = 1$ in the above equation gives the result that $y_1$ contains the most energy of any possible linear combination of the input data. Given this, setting $n = 2$ shows that $z_2$ contains as much energy as possible while remaining orthogonal to $z_1$. Etc. This result actually follows directly from the fact the $\sigma y_i^2$ are the eigenvalues of $\boldsymbol{R_x}$. For a full proof see e.g. Mardia [37].

Before the KLT can be applied, the covariance matrix of the input variables (image pixels) must be computed. There are two problems with this. First, as noted in the previous section, it is meaningless to talk about the covariance matrix for a *single* image; correlation is a statistical property which is only defined over an ensemble described by an image model. Second, even if such a model were readily available, the full covariance matrix between all pairs of pixels is far too large to be handled explicitly – and certainly finding $2^{16}$ eigenvectors of a matrix is a bit much to ask.

However, if the noise reduction algorithm to be applied involves processing the image in windows, both problems may be solved simultaneously. First, the set of all windows of a particular image can be considered to constitute samples from some underlying PDF. In other words, the underlying statistical distribution can be considered to be "all windows which look like the ones in this particular image". Second, the covariance matrix between all pixels within a window is small enough to be dealt with explicitly. The KLT can then be computed numerically. The window size must of course be chosen large enough that the pixel(s) at the center of the window which are to be estimated are essentially independent of pixels at the edges of the window (see section 9.1.)

One way to estimate the required correlations is simply to average the products between pixel values (including inter-channel products in the case of colour images) over all possible window positions. Note that this technique only yields correlation estimates for pixels which are both in the same window, which makes sense as windowed processing implicitly assumes that pixels not in the same window are statistically independent. Alternatively, spectral estimation techniques may be used to yield PSD and thence ACF estimates, from which the covariance matrix for the pixels within a window may be directly constructed.

After the covariance matrix is estimated by whatever means, performing a Karhunen-Loeve transform requires computing the eigenvectors of this matrix. For an $N \times M$ pixel window, this involves finding the eigenvectors of an $NM \times NM$ matrix, which is not a rapid operation, and produces a transformation matrix of the same size. During processing this transformation is applied to each window via matrix-vector multiplication, which requires $O(MN^2)$ operations. Compare this to the DFT, which may be implemented by a fast separable algorithm in $O(MN \log MN))$ time. Clearly, the full KLT is not a fast operation.

This problem is addressed by Jain [29]. First, if the ACF separable then the KLT will also separable. This immediately reduces the computational load, since the resulting horizontal and vertical covariance matrices will have dimension only $M \times M$ and $N \times N$ respectively. For this reason a separable transform is usually used. This appears to be an an adequate model of image statistics, for example noise reduction techniques based on separable transformations can be quite effective. However, from a drastically pragmatic point of view, if separable approximations

to the ACF are at all reasonable then they must be used, because the computational load of non-separable transformations is prohibitive.

The resulting separable KLT still requires the solution of an eigenvalue problem, and application of the resulting transformation by matrix-vector multiplication. It is for this reason that fast approximations to the KLT are desirable. First investigated in the course of research on image compression in the 1970s, the results in this field are now standard.

Real images tend to display an auto-correlation function which is well modeled by a function of the form $r(\boldsymbol{u}) = \exp(-a|\boldsymbol{u}|)$, i.e. exponential fall off with distance. This is precisely the correlation produced by a first-order Markov process, and the separable approximation to this radially symmetric function is fairly accurate [44]. Further, the Discrete Cosine Transform (DCT) is very close to the KLT where the covariance matrix is described by a first-order Markov process [31], and it can be computed in $O(N \log N)$ time. Thus the DCT is an excellent fast approximation to the KLT of image data. In fact the performance of the DCT on image data is virtually indistinguishable from the optimal KLT in terms of energy compaction; this is why it was chosen as the standard for the JPEG image compression algorithm.

For these reasons the DCT is quite frequently used in image processing applications which require decorrelation and/or energy compaction of the input data. For most images it is a very good approximation to the optimal KLT.

### 9.3.3 Feature Detection

All transformations designed to permit independent processing of transformed coefficients are in some sense attempting to construct separable approximations to the true joint PDF. Feature-based transformations differ from those examined previously in that they are imbued with prior knowledge of the important characteristics of the unknown joint density. Stated another way, feature detection techniques embody knowledge of the image model. The canonical example of this approach is the class of subband transforms, which are all based on the assumption that image features occur more or less independently at different scales and orientations.

In the feature-based approach, an image transformation is designed such that the basis functions into which the image is decomposed represent meaningful "features". In essence, by

specifying beforehand the building blocks which define meaningful image structure, the sub-optimal low-dimensional Bayesian estimation approximation is provided with additional a-priori information. This fact gives good reason to believe that a well chosen feature-based transform could out-perform any other transform technique.

There is another important reason for choosing feature-based transformations. Consider the error in the output image caused by severely mis-estimating a single transformed variable. This will result in an artifact in the shape of the basis function corresponding to the erroneous coefficient. This is a general principle which applies to any algorithm that employs image transforms: artifacts will always look like the basis functions used. For example, Fourier-based methods suffer from ringing, DCT methods suffer from the gridded patterns which are so characteristic of excessive JPEG compression, and wavelet methods produce odd "ripples" in the shape of the mother wavelet. The basis functions of feature-based transformations, on the other hand, may be selected in such a way as not to cause objectionable artifacts.

For these two reasons, transforms designed to detect image features are attractive. This is not a new idea, having been employed in the fields of computer vision and pattern analysis for some time (e.g. [19]). Even the classic Fourier-transform can be considered to represent the image by different classes of features, i.e. low and high frequency content. However truly effective feature based representations require the property of locality, otherwise the the non-stationary characteristics of real images cannot be modeled. This means that the transform basis functions must be localized in space and frequency or scale.

The idea of simultaneous localization in space and time is not new either, dating back to the pioneering work of Gabor on windowed Fourier transforms. It is safe to say however that the concept did not become widespread until the advent of wavelets, which are basis representations specifically designed to have certain localization properties.

However, because (orthogonal) wavelets involve critical downsampling of image data, they can suffer from aliasing artifacts. The Discrete Wavelet Transform (DWT) is of course perfectly invertible, but this is because the aliased terms of different sub-bands add perfectly to destructively cancel out all aliasing artifacts. This presents a problem if a single coefficient is disturbed slightly, as the aliasing will no longer exactly cancel, and will become visible. For

noise reduction, this problem is severe, because mis-estimation of a single transform coefficient – which will inevitably happen in practice – can cause very visible aliasing artifacts, even in otherwise smooth regions of the image. Another way of describing this problem is to note that wavelet basis functions are very high-frequency, and so must be perfectly balanced to represent smooth regions. Again, artifacts always look like the basis functions employed, and wavelets are highly "rough".

These and other issues relating to the lack of translation invariance and aliasing of wavelet transforms are discussed thoroughly by Simoncelli et. al. in [56]. The only solution is to avoid downsampling image bands which contain high-frequency data, resulting in an overcomplete representation which does not suffer from aliasing. Actually this idea goes back to the pioneering work of Burt and Adelson on pyramid-based image transformations [13], which are now ubiquitous. In terms of linear transformation, such transformations result in more transformed variables than input variables. Clearly then, some transform coefficients must be significantly correlated because there are more degrees of freedom in the transformed representation than the original image! Hence, this approach abandons all pretense of attempting to decorrelate the input values. Instead, it is hoped that the transformed coefficients provide a "meaningful" representation of the input data, perhaps representing concepts such as "there is an edge here".

Overcomplete pyramidal (or simply cascaded) subband image representations have been applied to noise reduction for some time. The canonical example is of course the subband coring algorithm of Adelson [1]. This basic algorithm has been subsequently re-examined in several studies such as Ranganath [45]. As a variation, instead of coring, the pixel-by-pixel Adaptive Wiener Filter of section 5.4.2 is applied to each level of the pyramid by Aiazzi et. al. [3].

Coring is actually a sort of "poor-man's Bayesian estimator" as discussed in section 8.5.2. The idea naturally arises of applying Bayesian coring to subband transforms, a technique which appears to have been first suggested by Simoncelli and Adelson in [55]. In that paper a significantly overcomplete transformation representation is used, containing sub-bands oriented for direction as well as scale information. This algorithm appears to represent the state of the art in noise reduction, however it has not been extended to colour images.

## 9.4 Phase Estimation

This section is something of a digression to explore a curious phenomenon and derive a very interesting result. This result is somewhat counterintuitive and is indicative, perhaps, of the art involved in designing dimension reduction strategies.

In section 5.5.1 a number of monochrome noise reduction techniques which rely on estimating only the magnitude of each frequency component of a signal were presented. In these algorithms the phase of the noisy signal is simply copied to the output, i.e. the observed noisy phase is used as an estimate for the phase of the original signal. Clearly if such an approach is warranted then the dimension of the estimation problem is immediately halved (assuming the use of Fourier-based techniques), because the phase of each frequency component represents exactly half of the image variables. Such techniques are referred to as *zero-phase* filters, and these will now be examined in detail.

### 9.4.1 Optimality of Zero-Phase Filters

It is the aim of this section to show under what conditions the observed noisy phase of a frequency component is the optimal estimator for the true phase.

Consider the complex amplitude $X_f$ of a single frequency component $f$ of the original noiseless signal. Due to the linearity of the Fourier transform, when noise is added to this signal the observed complex amplitude $Y_f$ at this frequency becomes

$$Y_f = X_f + N_f \tag{9.20}$$

where $N_f$ is the complex amplitude of the noise at frequency $f$. This situation is illustrated on a phasor diagram in figure 9.6. On this diagram, and henceforth, the phase of $X_f$ is assumed to be zero without loss of generality.

For any given noise amplitude $|N_f|$ the resulting noisy vector $Y_f$ may lie anywhere on the circle illustrated in the figure. Each such vector induces a phase error $\epsilon$ in the observed signal. The problem of phase estimation is then equivalent to estimating $\epsilon$. If one assumes that the noise phase $\theta_N = arg(N_f)$ is uniformly distributed from $0..2\pi$ then the error $\epsilon$ is symmetrically distributed around zero. If one further assumes that $X_f$ and $\theta_N$ are independent of all other

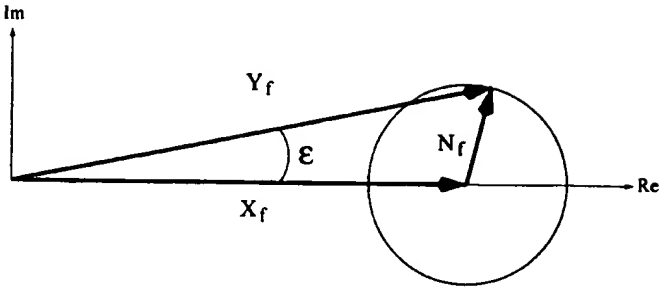



Figure 9.6: Phasor diagram showing the effect of noise on signal phase

variables, then the optimal estimate for $\epsilon$ in the Bayesian sense is clearly zero, corresponding to a zero-phase filter.

In the case of truly white noise, the phases of each component are indeed uniformly distributed over $0..2\pi$ and independent of one-another. These constraints also hold for any type of noise which can be obtained from white noise by application of a zero-phase filter, such as a convolution using a real-valued kernel.

However, real images display strong correlations between the phases of different frequency components, both within and between channels (for the case of colour images.) This must be so, or features such as sharp edges (which span all frequencies) would not be representable. Again, it is not clear why techniques which, in effect, only estimate the PSD of the signal should be effective.

### 9.4.2 Effects of Phase Errors

The question remains: why is it that using the noisy phase as an estimator of the true phase yields effective restoration techniques?

One possibility is that the phase is "unimportant" in some way. Yet, the phase information cannot be entirely discarded as the resulting phaseless image would be (by definition) just the autocorrelation function of the original signal, which rarely looks anything like a recognizable image.

Examining the effect of phase error on the squared-error measure sheds some light on the situation. Consider a signal $x$ and its accompanying Fourier transform $X(f)$. We can perturb

the phase of each component by an angle in $[-\epsilon, \epsilon]$ by multiplying $X$ by a function

$$P(f) = e^{ip_f} \tag{9.21}$$

where the $p_f$'s are independent uniformly distributed random variables in $[-\epsilon, \epsilon]$. The expected squared-error of this perturbed signal with respect to the original signal is

$$E = \mathrm{E}\left\{\int |(X(f) - P(f)X(f)|^2 \, df\right\} \tag{9.22}$$

Squared-error is traditionally defined in the spatial domain, but we have rewritten it above in the frequency domain, using basic properties of the Fourier transform (linearity and Parseval's theorem.) This equation can be further rewritten as

$$E = \mathrm{E}\left\{\int |(X(f)(1 - P(f))|^2 \, df\right\} \tag{9.23}$$

The expectation operator interchanges with integration due to its linearity and we may examine just the integrand at each frequency

$$e = \mathrm{E}\left\{|X(1 - P)|^2\right\} \tag{9.24}$$

Since, by assumption, the value of $P$ is independent of the value of $X$, the expectation integral is separable and we get

$$\begin{aligned} e &= \mathrm{E}\left\{|X|^2\right\} \mathrm{E}\left\{|(1 - P)|^2\right\} \\ &= \mathrm{E}\left\{|X|^2\right\} \mathrm{E}\left\{(1 - P)^*(1 - P)\right\} \end{aligned} \tag{9.25}$$

Substituting in equation 9.21 for $P$ and taking the expected value over the uniform distribution of error angles in $[-\epsilon, \epsilon]$ yields

$$\begin{aligned} e &= \frac{\mathrm{E}\left\{|X|^2\right\}}{2\epsilon} \int_{-\epsilon}^{\epsilon} (1 - e^{iq})(1 - e^{-iq}) \, dq \\ &= \frac{\mathrm{E}\left\{|X|^2\right\}}{2\epsilon} \int_{-\epsilon}^{\epsilon} 2 - 2\cos(q) \, dq \\ &= \mathrm{E}\left\{|X|^2\right\} \left(2 - 2\frac{\sin(\epsilon)}{\epsilon}\right) \end{aligned} \tag{9.26}$$

Since the term in parentheses is independent of frequency, the integral of equation 9.23 becomes

$$E = \left(2 - 2\frac{\sin(\epsilon)}{\epsilon}\right) \int \mathrm{E}\left\{|X(f)|^2\right\} \, df \tag{9.27}$$

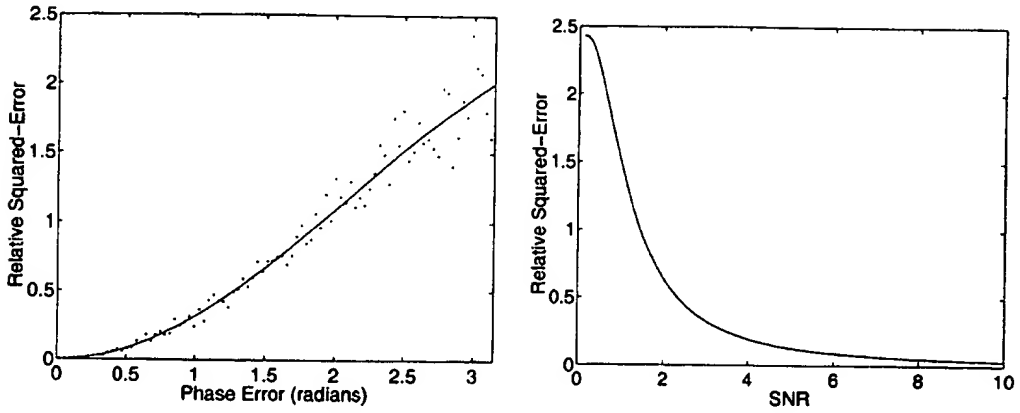



Figure 9.7: Left: expected relative squared-error as a function of maximum phase error, theoretical prediction and experimental results. Right: expected relative squared-error as a function of signal-to-noise ratio for an optimal zero-phase noise-reduction method.

and we see that the expected squared-error is a function of $\epsilon$ times the total signal power. The relative squared-error (defined as squared-error over signal power) is therefore just the function

$$r(\epsilon) = 2 - 2\,\mathrm{sinc}(\epsilon) \tag{9.28}$$

where $\mathrm{sinc}(x) \equiv \sin(x)/x$ as usual. This function is s plotted in figure 9.7.

This function has the properties we would expect: it goes to zero when the phase is unperturbed, and to one when $\epsilon = \pi$, corresponding to a total loss of phase information. It is also of course non-linear, and has an interesting property which finally explains why phase error can be tolerated: it is relatively flat for small values. For example, a phase error of $\pm 10°$ induces only 1% relative error in the resulting signal, and 10% relative error is not reached until the phase error is roughly $\pm 32°$.

As a test of the accuracy of this theoretical result, tests were run by perturbing the phase of the frequency components of a real image by varying amounts. The standard 'lena' image was used in grayscale form at a resolution of 256 by 256 pixels for these experiments. The results are shown as a scatter-plot on figure 9.7. The variance of experimental values around the theoretically predicted expected error gives some idea of the distribution of error values, which appears to be relatively narrow, especially for small perturbations, which is the region of interest in the evaluation of noise-reduction schemes.

We now examine the effect of the Signal-to-Noise ratio $\gamma$ on the phase error. Using the definitions of equation 9.20 the SNR (for a particular frequency) can be written as $\gamma = |X_f|/|N_f|$. From this value we wish to compute the maximum induced phase error. Referring to figure 9.6 this ocurrs when the noise is perpendicular to the signal, giving

$$\begin{aligned} \epsilon &= \arctan(|N_f|/|X_f|) \\ &= \arctan(1/\gamma) \end{aligned} \tag{9.29}$$

Srtictly speaking, we cannot compose this with equation 9.28 because that expression is derived under the assumption that the phase error is uniformly distributed in $[-\epsilon, \epsilon]$ whereas the expression above gives the *maximum* phase error of a non-uniform distribution. However, using the maximum error gives a reasonable approximation and will only overestimate the induced squared-error. Also, equation 9.28 was derived under the assumption that the phase error distribution is the same for all frequencies, whereas the SNR might vary over different parts of the spectrum (it is indeed precisely this phenomenon which the Wiener filter exploits.) However, we can again examine the worst case by taking the minimum SNR over all frequencies $\gamma_{min}$.

Under these approximations – which only increase the final error estimate and thus provide an upper bound – any technique which correctly estimates the magnitude of each frequency component but leaves the phase unchanged has an expected relative squared error of

$$r(\gamma_{min}) = 2 - 2\,\mathrm{sinc}(\arctan(1/\gamma_{min})) \tag{9.30}$$

This function is graphed in figure 9.7. Again, it has the expected properties: it increases rapidly as the SNR decreases, and goes asymptotically to zero as the SNR goes to infinity. The actual function values are also quite generous:

| SNR | Rel. SE |
|---|---|
| 2 | 0.636 |
| 5 | 0.126 |
| 10 | 0.033 |
| 15 | 0.015 |
| 20 | 0.008 |
| 30 | 0.004 |

For the case of film grain, referring back to chapter 3 and dividing pixel density by noise standard deviation, the SNR of digitized film ranges from a low of approximately 3 in the darkest regions of the image to a high of approximately 30 in the highest density areas. However grain noise variance falls off with the square root of density (as observed previously) and the second darkest gray step has an SNR of 13. Thus we come to the surprising result that using the noisy phase directly induces a relative error of about 1%, and the efficacy of zero-phase filters is explained. Should we wish to use Fourier-based techniques, this is an important optimization.

## 9.5 Colour is Irreducible

In section 6.1 techniques designed to allow independent processing of the channels of a colour image were investigated at length. Based on this examination and subsequent experiments, it did not seem likely that any sort of independent channel processing would be effective for noise reduction in colour images. It is the aim of this section to to prove that this is so, investigate why, and to suggest low-dimensional techniques which might be suitable for near-optimal multi-channel estimation.

In the optimal Bayesian estimator sense, independent processing of each channel is possible only if the joint PDF across all pixels and channels is separable, that is

$$p(\boldsymbol{r}, \boldsymbol{g}, \boldsymbol{b}) = p_r(\boldsymbol{r}) p_g(\boldsymbol{g}) p_b(\boldsymbol{b}) \tag{9.31}$$

where $\boldsymbol{r}$, $\boldsymbol{g}$, and $\boldsymbol{b}$ are the values of all pixels in the red, green, and blue channels respectively.

As written, equation 9.31 is hard to verify directly, due to the large number of variables involved. Instead, since we are interested in the negative result, it suffices to disprove any necessary condition. Integrating both sides of 9.31 over all pixel positions gives the corollary

$$p(r, g, b) = p_r(r) p_g(g) p_b(b) \tag{9.32}$$

where $r$, $g$, and $b$ are now taken at a *single* pixel. If it is the case that this condition does not hold, then the channels of an image are not independent and any optimal algorithm must consider all channels at once. If it can further be shown that there is no good separable

approximation to the joint PDF $p(r, g, b)$ then we have the stronger result that no independent channel algorithm can be a good approximation to the optimal estimator.

Thus we are led to examine the distribution of colour values in real images. Figure 9.8 displays a scatter plot of 4000 RGB pixel values obtained randomly from an ensemble of over 100 different digitized photographs of various subjects.

This distribution displays a lot of structure. First, note that it is clustered near the central axis of the RGB colour cube, which runs from black to white. This is because natural scenes tend not to contain extremely saturated colours. Second, there is a very distinct line exactly along the central axis, indicating the predominance of almost pure greys. There are also strong clusters near both the black and white points. This may either indicate that many pixels are found in the extremes of the tonal range, or that the scale chosen for digitization of intensity values has insufficient range to capture the full dynamic range of the original scenes, causing clipping of very bright or dark colours. Examining the overall shape of the distribution, it is clear that saturated colours occur far most often in the mid-tones. Also, this distribution is somewhat skewed towards the reds. The reason for this is unclear. It may be due to colour correction of the images in the test ensemble, or properties of the spectral response of colour film, or maybe it is due to a predominance of skin tones in the input images.

At any rate, this distribution has complex features and is certainly not well modeled by a simple multi-variate Gaussian. It does however have an obvious "principal axis". In fact, the shape of this distribution explains why the Karhunen-Loeve transform applied to decorrelate the colour channels is able to concentrate so much energy into one channel. Figure 9.9 shows the same distribution after application of the Karhunen-Loeve transform. (The sharp cut off of intensity values at $\sqrt{3}$ represents the fact that each of RGB component has a maximum of 1.) It is clear that the KLT simply effects a rotation of the colour coordinates. As expected, the black-white axis of RGB space is aligned with one of the variables, here labeled "Intensity", while the perpendicular directions represent the colour information and are therefore labeled "Chroma 1" and "Chroma 2". While most of the variance is occurs along the intensity axis, this distribution is truly three dimensional in the sense that discarding any dimension — even if it is in the direction of minimum variance – will have drastic consequences.

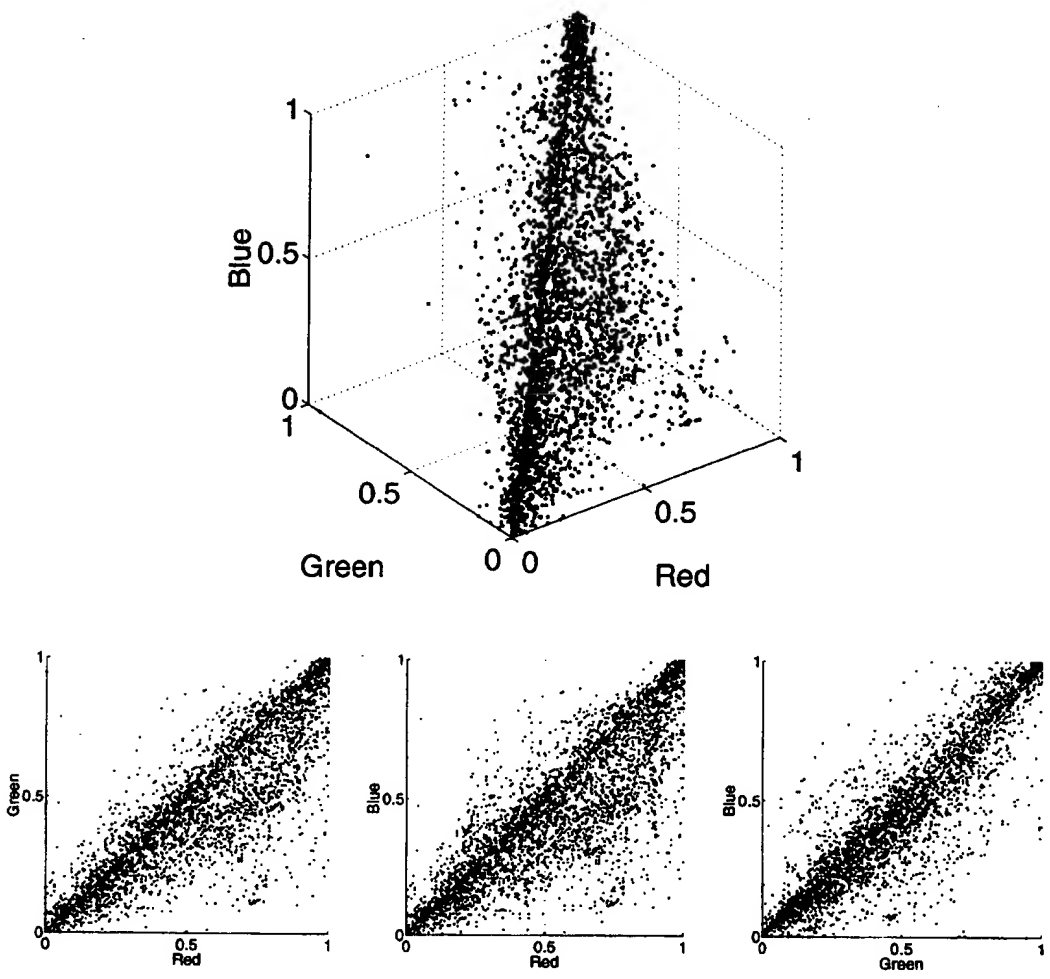



Figure 9.8: Distribution of RGB pixel values obtained from real photographic images, oblique view and orthogonal projections.

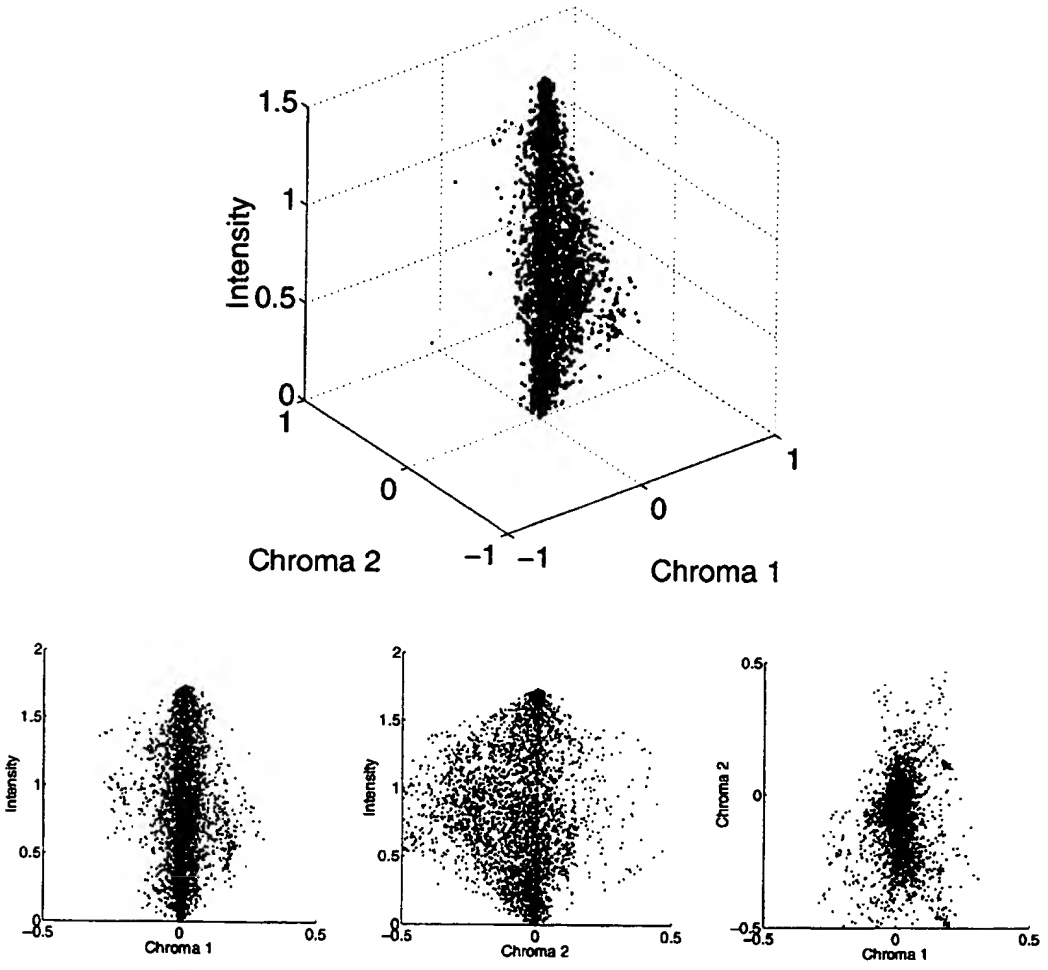



Figure 9.9: The same pixel value samples as figure 9.8 after transformation into Karhunen-Loeve colour space.

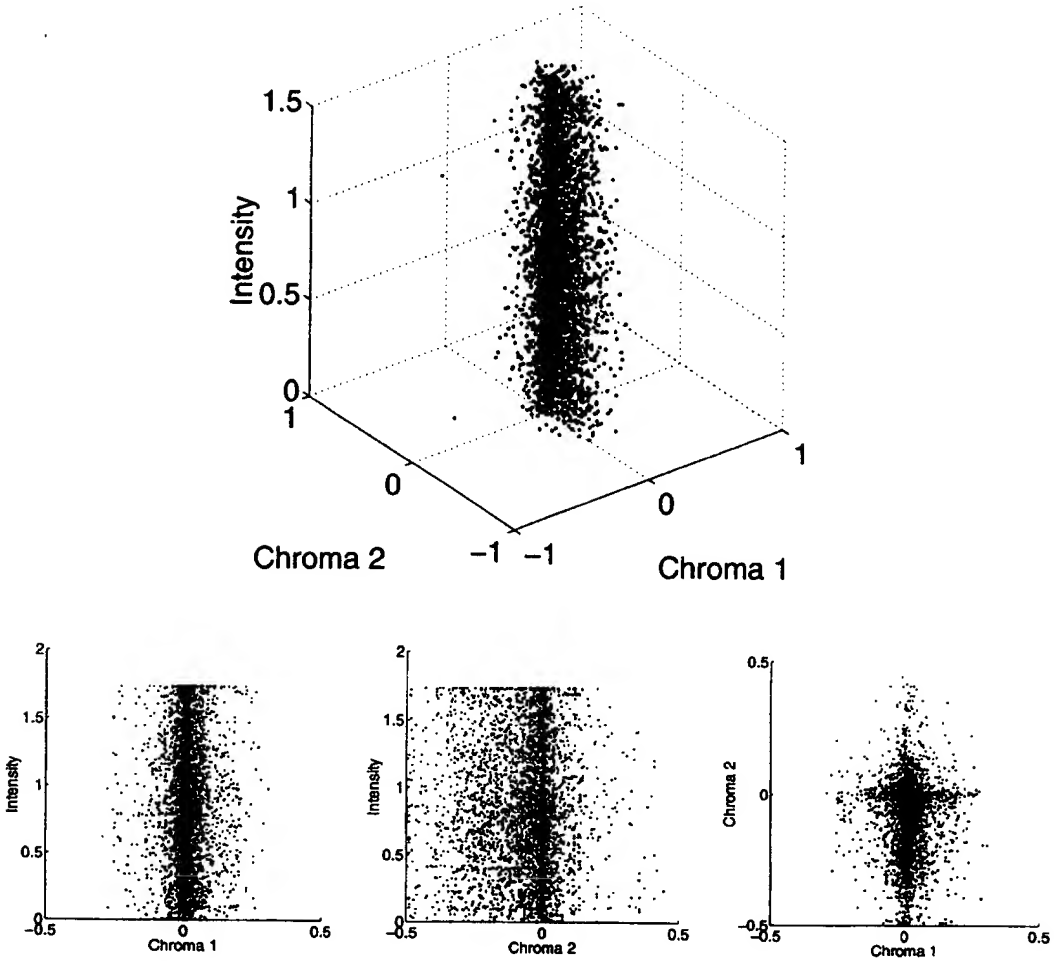



Figure 9.10: Separable approximation to the distribution of pixel values in KL colour space, obtained by independent sampling of the marginal probabilities along each axis.

If the transformed image channels are estimated separately, the effect is as if the true distribution is replaced by the product of the marginal densities. Figure 9.10 shows a visualization of the resulting separable approximation. The points in this scatter plot were obtained by independent re-sampling along each axis in KL colour space. Many important features of the original distribution are lost. Most notably, this separable approximation lacks the characteristic bulge along the chromatic directions which occurs for medium intensities. Thus the fact that saturated colours do not tend to be very light or dark is not represented in this distribution. Also, the clusters near black and white are similarly spread out. In the chromatic plane, pixel values are now far more likely to lie close to one of the coordinate axes. Since rotation on this plane corresponds to changes in hue, clustering in certain directions means that the distribution displays preferences for particular hues.

This simple demonstration shows why colour-space transformation is no substitute for true multi-channel estimation. Basically, the shape of the distribution of colour values just cannot be well approximated by a product of marginal densities, even in a colour-space where the colour coordinates are completely uncorrelated.

It is now clear that the image channels must be processed together at some stage. One possibility is to derive an estimator which operates between the channels of each pixel individually; such a technique is certainly expressible in the Bayesian estimation framework, and the distribution of pixel values is exactly the a-priori information needed for such an approach. However, this approach does not use any spatial correlation information at all, so it must perform very poorly. A simple "fix" is to perform such cross-channel estimation before or after the application of a monochrome estimation technique to each channel independently. This is certainly feasible but it is somewhat dubious from a theoretical standpoint; one major problem with this approach is that dependencies across both the spectral and spatial dimension *simultaneously* are ignored. Instead, it seems that a fully cross-channel solution is required. The exact form which such a solution might take is discussed next.

## 9.6 Separable Multi-channel Model

Based on the analysis of the previous section it is clear that the spectral distribution $p(r, g, b)$ is not separable. In contrast, as discussed in as in section 9.3, single channel noise reduction techniques can get away with estimating only one variable at a time given the correct choice of transformed domain. These results can be combined to yield a form which appears to be a good separable approximation to the full multi-channel posterior distribution.

The efficacy of transform methods implies that the distribution which describes the spatial interdependencies of pixels within a single channel is often well approximated by a separable function along some set of axis, as described by equation 9.10. Writing this equation for each channel independently gives the marginal distributions

$$\begin{aligned} p(\boldsymbol{r}') &\approx p_{r_1}(r'_1)p_{r_2}(r'_2)\dots p_{r_n}(r'_n) \\ p(\boldsymbol{g}') &\approx p_{g_1}(g'_1)p_{g_2}(g'_2)\dots p_{g_n}(g'_n) \\ p(\boldsymbol{b}') &\approx p_{b_1}(b'_1)p_{b_2}(b'_2)\dots p_{b_n}(b'_n) \end{aligned} \tag{9.33}$$

where $\boldsymbol{r}' = [r'_1\, r'_2 \dots r'_n]^T$ etc. and the primes denote that all channels have been spatially transformed in an identical manner, i.e.

$$\begin{aligned} \boldsymbol{r}' &= \boldsymbol{A}\boldsymbol{r} \\ \boldsymbol{g}' &= \boldsymbol{A}\boldsymbol{g} \\ \boldsymbol{b}' &= \boldsymbol{A}\boldsymbol{b} \end{aligned} \tag{9.34}$$

for some linear transformation $\boldsymbol{A}$. These equations appear to be good models for each channel taken independently, assuming a suitable choice of $\boldsymbol{A}$. However the full multi-channel joint distribution $p(\boldsymbol{r}, \boldsymbol{g}, \boldsymbol{b})$ is not separable across channels in any fashion. Combining these two facts we may postulate a factoring for the multi-channel PDF as

$$p(\boldsymbol{r}', \boldsymbol{g}', \boldsymbol{b}') \approx p_1(r'_1, g'_1, b'_1)p_2(r'_2, g'_2, b'_2)\dots p_n(r'_n, g'_n, b'_n) \tag{9.35}$$

This equation describes a joint density which is not spectrally decomposable in any way, but which is well approximated by a product with one factor per coefficient in some spatially transformed domain. Both of these conditions seem to hold in practice: this section has shown

that RGB data is not separable, and the existence of effective single-channel techniques (e.g. spectral subtraction, subband coring) suggests that the transform domain factoring described by equation 9.33 is achievable. Thus there is good reason to believe that 9.35 is an accurate description of the multi-channel PDF (with suitable choice of spatial transformation $\boldsymbol{A}$ ).

An algorithm is immediately suggested. First, all image channels are spatially transformed, independently, in the same manner. Then, each transform coefficient is estimated independently across all three channels. The resulting estimator is three-dimensional. If 9.35 holds, such an algorithm may yield very good results.

Note that because all variables of equation 9.35 are spatially transformed, the equation is not symmetric in spatial and spectral variables. The image must be spatially transformed *first* and then estimated across channels. The reverse ordering, where each pixel is estimated across channels and then each channel is independently transformed, will be significantly sub-optimal because it ignores the strong spatial-spectral correlations discussed in the previous section.

In summary, there seems absolutely no way to get around the fact that colour is a three-dimensional quantity. This puts a lower bound of three on the minimum dimension of the estimation integral of any truly multi-channel noise reduction algorithm. On the bright side, it seems likely that this bound can be achieved, given the multi-channel joint PDF model of equation 9.35.

# Chapter 10

# Multi-Channel Bayesian Coring

In chapter 8 it was argued that Bayesian estimation provides a unified framework for noise reduction, but it was also shown that brute force application of this technique requires probability distributions which have hundreds of thousands of variables. In chapter 9 various techniques for reducing this dimension by replacing the full PDF with separable approximations were presented, and the implications for multi-channel images were explored.

In particular, an effective approximation to the full high-dimensional distribution of "real colour images" was postulated in equation 9.35, repeated here:

$$p(r', g', b') \approx p_1(r'_1, g'_1, b'_1)p_2(r'_2, g'_2, b'_2)\ldots p_n(r'_n, g'_n, b'_n) \tag{10.1}$$

with

$$\begin{aligned} r' &= Ar \\ g' &= Ag \\ b' &= Ab \end{aligned} \tag{10.2}$$

where $r$, $g$, and $g$ are the stacked vector representations of the red, green, and blue channels respectively and $A$ is some linear transformation. Arguments were presented that such a model might in fact be accurate enough to allow effective multi-channel noise reduction.

This chapter presents one possible algorithm based on this model. The resulting technique is shown to be more effective in removing film grain than any existing single or multi-channel technique.

## 10.1 Choice of Linear Transform

There are any number of possible ways to implement a noise reduction system based around equations 10.1 and 10.2. After application of the dimension reducing transformation to each channel independently, each transform coefficient is simultaneously estimated across all three channels.: Bayesian estimation of a three-element vector is performed, using three-dimensional probability distributions. Possible transform choices include:

- global application of a DFT,
- block processing using the DFT or DCT,
- wavelet transforms, and
- subband transforms.

Each of these transforms has its advantages and disadvantages. Global transforms (such as a global Fourier Transform) cannot adapt to differences in signal statistics within the same image, and thus suffer from artifacts such as ringing. The remaining choices above all employ some sort of local transformation and thus do not suffer from such problems. Block processing can be effective but choosing a block size is difficult, and such transforms can suffer from a lack of translation invariance. Wavelet transforms suffer from severe translation artifacts, fundamentally this is related to the aliasing which occurs in the downsampling stage as discussed in section 5.7.

This leaves overcomplete subband transforms, which were studied during the discussion of subband coring techniques in section 5.6. For monochrome images the approach of subband coring provides fairly effective noise reduction, but there is no obvious generalization to multi-channel images. However, if coring is re-interpreted in a Bayesian estimation framework, as was done in section 8.5.2, then the multi-channel extension becomes clear: the colour coring process can be effected by three-dimensional Bayesian estimation.

### 10.1.1 Subband Filter Design

The construction of a separable 2D subband transform from a single pair of 1D band splitting filters was discussed in section 5.6.2. The principal feature of such a transform is the cascaded partitioning of frequency space by recursively partitioning the low-pass subband.

Ideally, since each subband has a narrower bandwidth than the original signal, it should be representable with fewer samples. This raises aliasing issues relating to the downsampling of subbands and, as discussed in section 5.6.2, in general a subband transform which is useful for noise reduction must be overcomplete to prevent such aliasing. However, it was also noted that if the low-pass subband at each stage does not have appreciable energy above 0.25 cycles/pixel, it may be decimated by a factor of two without introducing aliasing and later accurately reconstructed. Repeating this downsampling procedure at each stage of a cascaded transform produces a pyramidal subband transform, where each successive level is half the size of the previous one, a process illustrated schematically in figure 5.7. This property greatly reduces processing time required both to perform the subband transform and to operate on the resulting coefficients, thus it is highly desirable for a practical noise reduction system.

The low-pass kernel of the filter pair presented in section 5.6.2 has significant power above 0.25 cycles/pixel, so the resulting subband transform used for illustration and testing purposes in chapter 5 was not downsampled. The transforms used in Adelson's original coring algorithm [1, 14] were not downsampled either. Nor does there seem to be any existing work on developing separable subband transforms which can be downsampled. Later work by Simoncelli [56] involves the construction of a non-separable transform where the low-pass filter has sufficiently sharp cutoff characteristics to allow subsampling. However the actual numeric filter coefficients are not reported. Therefore, one way or another, a subband transform had to be designed before the proposed multi-channel algorithm could be implemented.

Although in principle non-separable kernels offer many advantages including better directional sensitivity and discrimination between the two diagonal orientations, for preliminary experiments it was felt that a separable transform was sufficient. Further, separable transforms are considerably faster to compute, and much simpler to design and implement. Of course, it may very well be that non-separable kernels offer an advantage for noise reduction. This

proposition must be verified in future work.

In short, a subband transform with the following properties is desired:

- Self inversion,
- Efficient implementation via small separable filters,
- pyramidal structure,
- non-aliased subbands

In turn, these requirements motivate the construction of a 1D low-pass/high-pass filter pair, to be used separably in a pyramidal subband system, with the following properties:

- The filter pair must satisfy, as closely as possible, the perfect reconstruction condition of equation 5.49 to allow self-inversion.
- Both filters should have as few taps as possible.
- The low-pass filter must have negligible power beyond 0.25 cycles/pixel.

In addition, if the low-pass filter is also to be used as the interpolation filter during the upsampling required in the reconstruction process (as discussed in section 5.6.2) the sum of its even and odd coefficients must be equal. Otherwise, upsampled images will display a gridded appearance as the filter alternates between different placements with respect to non-zero samples. Actually, this requirement is equivalent to the condition that the frequency response of the low-pass kernel be identically zero at the Nyquist limit of 0.5 cycles/pixel, because the highest frequency term in the Fourier transform of a discrete sequence is just the difference between the sums of the even and odd samples. If frequency content at 0.5 cycles/pixel is not suppressed, the reconstructed signal will exhibit fluctuation between even and odd pixels. Figure 10.1.1 illustrates this problem. Of course, ideally the low-pass filter will completely eliminate all frequencies above 0.25 cycles/pixel anyway, but this property is not obtainable with a finite filter length, so "even-odd symmetry" must be explicitly enforced.

Similarly, the coefficients of the high-pass filter must sum to zero, to ensure that the DC component of the filtered signal is completely blocked, and the coefficients of the low-pass filter

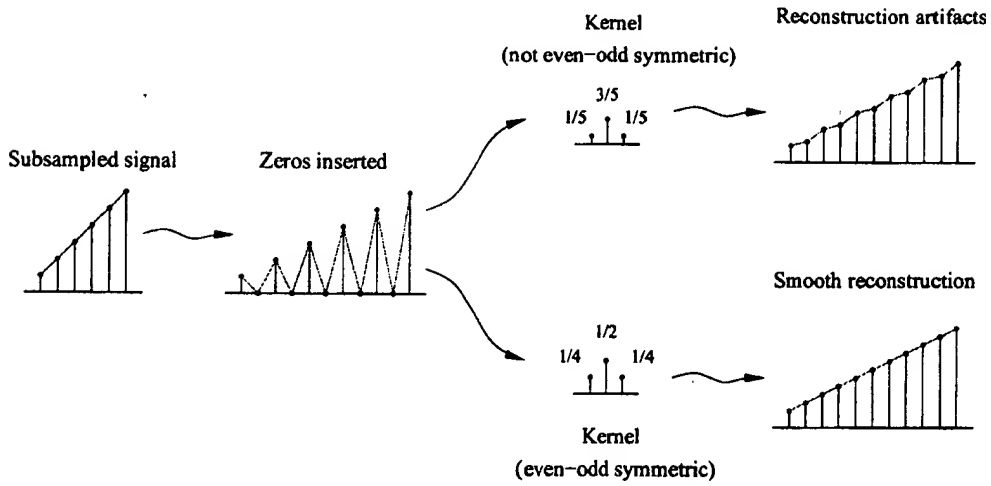



Figure 10.1: Demonstration of the need for reconstruction kernels which have equally weighted even and odd coefficients. If the kernel employed is not even-odd symmetric in this fashion, the reconstructed signal will fluctuate as the filter shifts between the two possible phases with respect to the non-zero coefficients of the zero-padded intermediate signal.

must sum to one, to ensure that the overall signal level is unchanged. Again, these properties would be automatically satisfied by an ideal band-splitting filter pair, but must be explicitly enforced when constructing finite-length approximations.

A filter pair satisfying all the above constraints was constructed by numerical optimization. First, the filter length for both kernels was arbitrarily set to 9 coefficients. This length was chosen as a reasonable tradeoff between computational efficiency and filter optimality, based on previous experience with filter design. Note however that symmetry reduces the number of free variables to 5 for each filter. The resulting ten parameters were optimized numerically subject to a cost function inspired by [54], having the following terms:

$$re = \int_0^{1/2} (|L(f)|^2 + |H(f)|^2 - 1)^2 df \tag{10.3}$$

$$de = \int_0^{\omega_c} (|L(f)|^2 - 1)^2 df + \int_{1/4}^{1/2} |L(f)|^2 df + \int_0^{\omega_c} (|H(f)|^2)^2 df + \int_{1/4}^{1/2} (|H(f)|^2 - 1)^2 df \tag{10.4}$$

$$ne = \left|\sum_i l_i - 1\right| + \left|\sum_i h_i\right| + \left|\sum_{i\ \text{even}} l_i - \sum_{i\ \text{odd}} l_i\right| \tag{10.5}$$

where $l_i$ and $h_i$ are the filter coefficients of the low-pass and high pass kernels, respectively, and $L(f)$ and $H(f)$ are the Fourier transforms of these two kernels. The *re* term ("reconstruction error") penalizes violation of the self-inversion condition of equation 5.49, while the *de* term ("design error") penalizes deviations from the desired filter design. The first two terms of *de* force the low-pass filter to have unity response from DC to a cutoff frequency of $\omega_c$, which defines the start of the transition band, and zero response from 0.25 cycles/pixel upward. It is this latter condition which allows the low-pass subband to be downsampled without aliasing. The remaining terms of *de* enforce similar design constraints on the high-pass band. All frequency domain integrals were evaluated by numerical integration using simple quadrature over a relatively small number of points; this is effective because the power spectra of short kernels are quite smooth. Finally, the *ne* term ("normalization error") ensures that the low-pass coefficients sum to one, the high pass coefficients sum to zero, and that the sums of the even and off coefficients of the low-pass filter are equal.

Note that some of the conditions imposed by the *de* and *ne* terms are redundant when combined with the perfect reconstruction condition embodied in the *re* term, however the addition of these terms helps to direct the numerical optimizer away from unwanted local minima in the design space.

The final cost function was set to

$$c = \alpha\, re + (1 - \alpha)\, de + ne \tag{10.6}$$

where $\alpha \in (0..1)$ controls the relative importance of reconstruction error versus satisfaction of the filter design constraints. This cost function was numerically minimized over the ten free parameters by a conjugate directions line-search method from MATLAB's optimization toolkit. After some experimentation the values $\alpha = 0.9$ and $\omega_c = 0.01$ (cycles/pixel) were settled upon as providing a reasonable tradeoff between reconstruction error and filter properties. The resulting filter pair is given in table 10.1.1.

The responses of thesse two filters are plotted in figure 10.2. Note that to satisfy the hard constraint that the low-pass filter have negligible response above 0.25 cycles/pixel, the

| Low-Pass | High-Pass |
|---|---|
| -0.03271 | -0.01386 |
| -0.01241 | -0.05609 |
| 0.11169 | -0.12523 |
| 0.26240 | -0.19354 |
| 0.34206 | 0.77743 |
| 0.26240 | -0.19354 |
| 0.11169 | -0.12523 |
| -0.01241 | -0.05609 |
| -0.03271 | -0.01386 |

Table 10.1: The coefficients of the filter pair designed in this section.

reconstruction property has been sacrificed somewhat, though tests show that the average relative reconstruction error at each pixel is on the order of 0.5% . Also, the transition band is quite wide. These problems could be alleviated through the use of wider filter kernels.

The resulting cascaded subband system is shown schematically in figure 10.3. Each separable filtering operation is denoted by $a, b$ where $a$ is the filter applied in the horizontal direction and $b$ is the filter applied in the vertical direction, and each filter is one of $l$ (denoting the low pass filter) or $h$ (denoting the high-pass filter). The input signal $x$ is decomposed into horizontal, vertical, and diagonal bands at each scale, plus a residual low-pass image at the final level of cascading. The three level decomposition of the Lena test image using this system is shown in figure 10.4.

## 10.2 Multi-Channel Bayesian Estimation

From equation 8.5.2 the minimum variance estimator (of all pixels and all channels simultaneously) is known to be

$$\tilde{\boldsymbol{x}} = \int \boldsymbol{x} \frac{p_{\boldsymbol{x}}(\boldsymbol{x}) p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y} \,|\, \boldsymbol{x})}{p_{\boldsymbol{y}}(\boldsymbol{y})} \, d\boldsymbol{x} \tag{10.7}$$

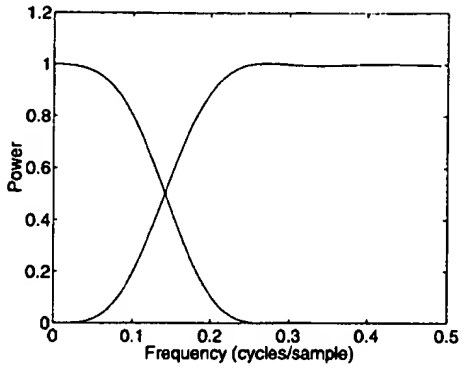



Figure 10.2: Frequency response of the 9-tap band splitting filter pair designed in this section.

where $p_x(\boldsymbol{x})$ is the probability that the original uncorrupted image is equal to $\boldsymbol{x}$ (the prior probability), $p_{y|x}(\boldsymbol{y}|\boldsymbol{x})$ is the probability that the noisy image $\boldsymbol{y}$ will be observed if the original image is $\boldsymbol{x}$, and $p_y(\boldsymbol{y})$ is the probability that $\boldsymbol{y}$ is observed regardless of the value of $\boldsymbol{x}$ (the Bayesian evidence). If we further assume the usual additive noise model $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ then this simplifies to

$$\tilde{\boldsymbol{x}} = \frac{\int \boldsymbol{x} p_x(\boldsymbol{x}) p_n(\boldsymbol{y} - \boldsymbol{x})\, d\boldsymbol{x}}{p_y(\boldsymbol{y})} \tag{10.8}$$

where $p_n$ is the probability distribution of (all pixels and all channels of) the noise. If the crucial separable approximation of equation 10.1 holds, then equation 10.8 can be applied to the three colour channels each subband coefficient independently; that is, all vectors and distributions in equation 10.8 become three-dimensional.

The Bayesian Evidence $p_y$ may be computed as

$$p_y(\boldsymbol{y}) = \int p_x(\boldsymbol{x}) p_n(\boldsymbol{y} - \boldsymbol{x})\, d\boldsymbol{x}. \tag{10.9}$$

Therefore, evaluation of 10.8 requires knowledge of the subband coefficient distribution $p_x$ and the noise distribution $p_n$. Note that these distributions will in general be different for each subband.

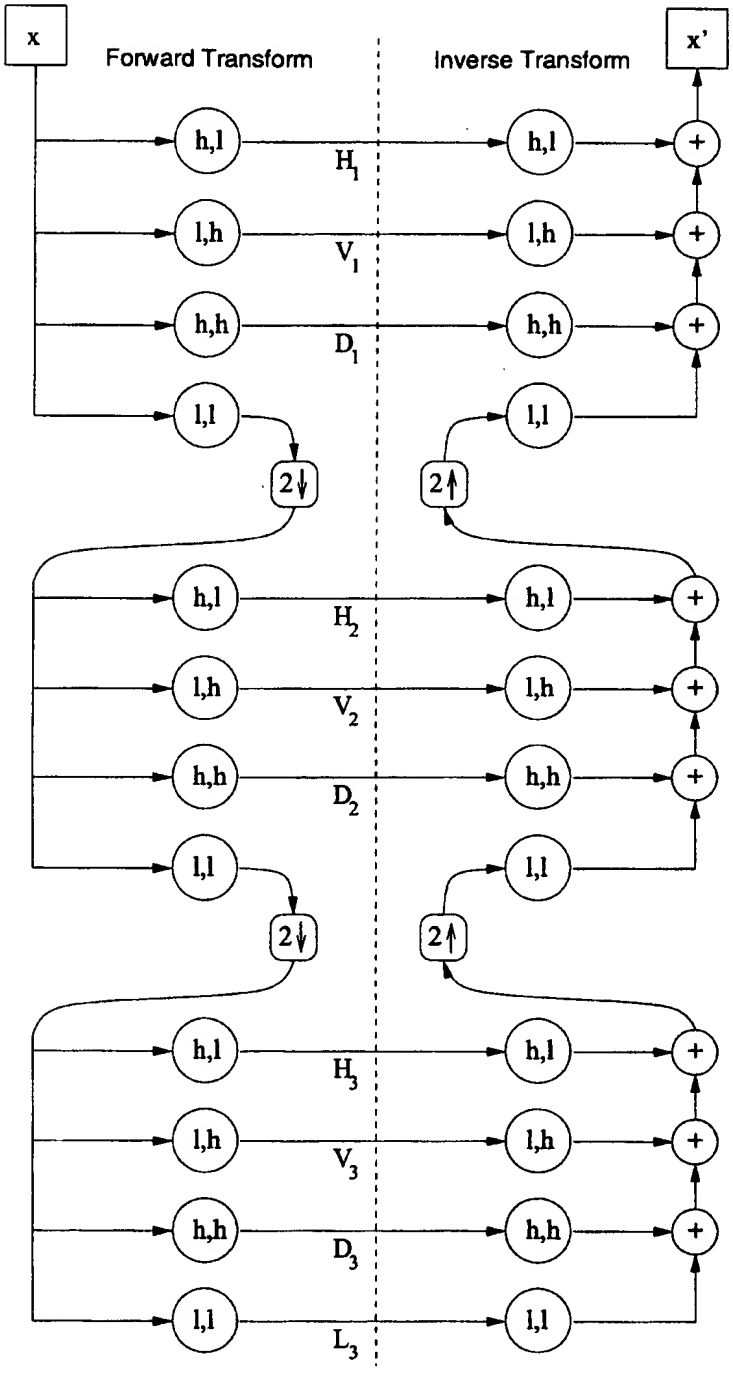



Figure 10.3: Structure of the pyramidal subband transform built using the filter pair derived in this section. Two levels of recursion are shown.

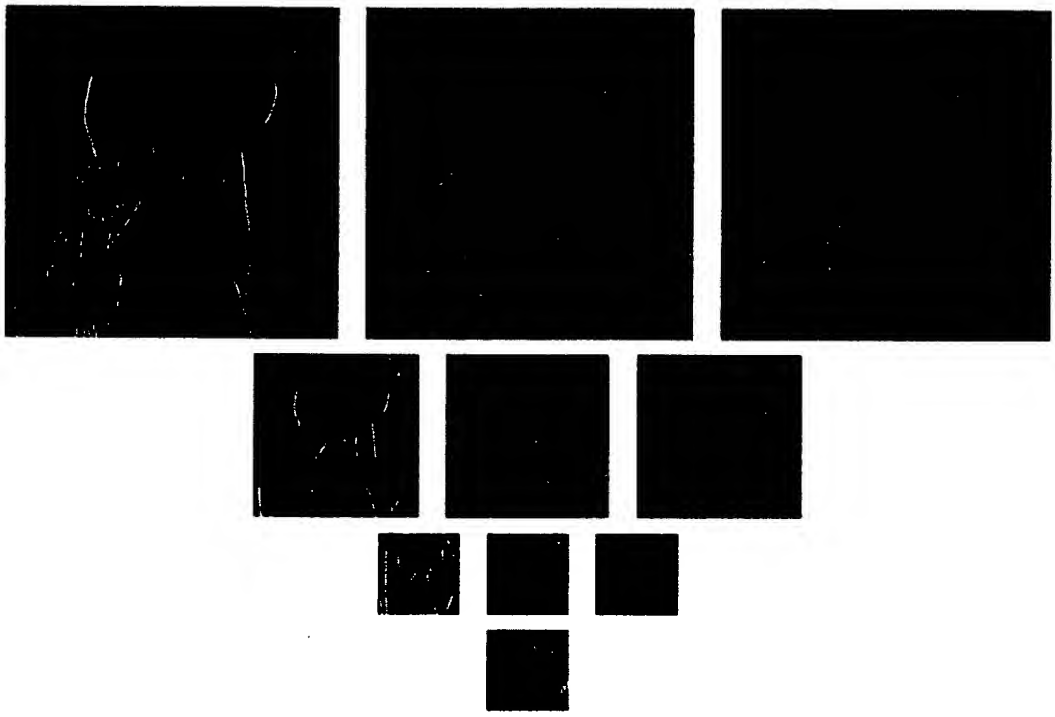


Figure 10.4: Three level subband decomposition of Lena.

### 10.2.1 Noise Distribution

Chapter 3 explored grain noise and found it to be Gaussian, signal-dependent, non-white, and independent between channels. Chapter 4 discussed techniques for extracting samples of the noise and estimating the noise PSD.

Using the fact that grain noise is independent between channels, the multi-channel noise distribution can be written, across all pixels as

$$p_n(r, g, b) = p_{n_r}(r)p_{n_g}(g)p_{n_b}(gb). \tag{10.10}$$

Note that this expression cannot be written for each pixel individually because the non-whiteness of the noise implies dependencies between pixels. However channel independence does allow us to study and develop a simple single channel noise model. In the remainder of this section, a monochrome image is assumed.

Any linear combination of Gaussian random variables is also a Gaussian random variable [38], so after subband transformation each transform coefficient will also display Gaussian noise. Note however that this noise will be correlated between different coefficients in the same subband, i.e. the noise will not be white. This is to be expected: after all, a subband transform (ideally) suppresses all frequencies except for a narrow band, so it cannot ever produce white noise regardless of the characteristics of the input signal. This is another manifestation of a phenomenon encountered during the study of colour-space transformations in section 6.1, where it was shown that because noise is independent between channels in the RGB system, any colour-space transformation which decorrelates the image between channels must correlate the noise between channels. Analogously, since the range of noise correlation is small and the range of signal correlation is large, any transform which effectively detects image features cannot produce noise which is independent at each pixel.

Unfortunately, operating on each subband coefficient independently means that correlations between multiple coefficients cannot be taken into account. Essentially, the estimation process intrinsically assumes that the noise is white. This does not mean that the entire noise reduction process acts as if the noise is white. On the contrary, each subband can be processed under the assumption of a different (white) noise variance. One can think of the true noise PSD as being

approximated by a piecewise constant function over the frequency plane, split according to the subband regions. Of course, this analogy is not strictly correct because of the wide transition regions between subbands, but it does serve to illustrate the fundamental space/frequency resolution tradeoff: spatially compact kernels must necessarily have poorer frequency discrimination than a Fourier transform, thus no spatially compact transformation can utilize a noise PSD estimate at full resolution.

Given this analysis, the noise modeling task is reduced to computing the variance of the noise in each subband. Note that, if the noise is assumed to have stationary statistics and each subband is generated by convolution with a space-invariant kernel, then this variance will be the same for every coefficient belonging to a particular subband.

Since subband transforms are linear, it suffices to investigate the effect of such transforms on noise alone. Each resulting subband coefficient $t$ is a linear combination $\boldsymbol{B}$ of a finite set of pixels, each of which contains Gaussian noise. Let the noise values in this set of pixel be denoted $\boldsymbol{n}$. Then,

$$t = \boldsymbol{B}\boldsymbol{n}. \tag{10.11}$$

(Note that since $t$ is a scalar, $\boldsymbol{B}$ is a row vector rather than a full matrix.) We wish to know the variance of $t$. If all elements of $\boldsymbol{n}$ are zero-mean, then so is $t$, so this variance is simply

$$\mathrm{E}\left\{t^2\right\} = \mathrm{E}\left\{\boldsymbol{B}\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{B}^T\right\}. \tag{10.12}$$

But, $\mathrm{E}\left\{\boldsymbol{n}\boldsymbol{n}^T\right\}$ is just the covariance matrix $\boldsymbol{R}_{\boldsymbol{n}}$ of the the noise in the pixels of $\boldsymbol{n}$, so

$$\mathrm{E}\left\{t^2\right\} = \boldsymbol{B}\boldsymbol{R}_{\boldsymbol{n}}\boldsymbol{B} \tag{10.13}$$

By assumption the noise has stationary statistics, producing a well-defined PSD. Knowledge of the noise PSD defines the noise auto-correlation function, which describes the correlation between the noise values in any two pixels as a function only of the distance between these pixels. Thus, $\boldsymbol{R}_{\boldsymbol{n}}$ is a (block) Toeplitz matrix constructed from the elements of the noise ACF.

This fact enables direct computation of the noise variance for each subband using equation 10.13, given the noise ACF and subband kernel coefficients. While suitable for the high-frequency subband levels, this direct approach is not effective for the lower frequency bands.

The problem is that $\boldsymbol{B}$ must take into account all pixels which are used in computing each subband. The use of pyramidal systems allow very large effective kernel sizes at low computational cost, but equation 10.13 requires that every pixel which contributes to the subband coefficient be taken into account, so the $\boldsymbol{B}$ must explicitly represent the *full* effective kernel. Further, separability cannot be exploited in equation 10.13 so these effective kernels must represent the actual two dimensional pattern of pixels contributing to each coefficient. Because of the downsampling and repeated convolution employed at each level, the effective kernel size increases exponentially at each successive level of the transform pyramid. For the 9-tap filters constructed earlier, the third-level effective kernels will have size $57 \times 57$, encompassing $57^2 = 3249$ pixels. The corresponding correlation matrix $\boldsymbol{R_n}$ would have $3249^2 = 10,556,001$ elements. Clearly, direct use of equation 10.13 is expensive at best and infeasible at worst.

Fortunately, because grain noise is typically appreciably correlated over only a few pixels, the corresponding ACF has very small support. Thus the vast majority of pixels encompassed by a large effective kernel are completely independent. Correspondingly, $\boldsymbol{R_n}$ is extremely sparse. This makes efficient storage and multiplication possible, and explicit implementation of sparse block-Toeplitz matrix routines is certainly a possibility. However, this task may be avoided entirely by use of the formal analogy between multiplication by Toeplitz matrices and discrete convolution. That is, a block-Toeplitz matrix exactly represents the application of a two-dimensional convolution operation where edge effects are handled by padding the signal with zeros. A square block-Toeplitz matrix further represents the case where only the central portion of the convolution which is the same size as the input signal is returned. Therefore,

$$\boldsymbol{R_n B}^T = \mathit{stack}(\boldsymbol{A_n} \star \boldsymbol{K_B}) \tag{10.14}$$

where $\boldsymbol{A_n}$ is the 2D noise ACF, $\boldsymbol{K_B}$ is the 2D convolution kernel corresponding to the linear combination $\boldsymbol{B}$, and *stack* is the usual 2D stacking operator which encodes pixel arrays into vectors. The convolution in this equation is taken to extend the kernel $\boldsymbol{K_B}$ with zeros as needed, and return a result only as large as $\boldsymbol{K_B}$. Since the noise ACF is usually very small, this convolution can be computed very efficiently. Also, if the effective kernel $\boldsymbol{K_B}$ is separable, this convolution can also be performed separably.

Combining equations 10.13 and 10.14 produces the following algorithm:

1. Construct the effective horizontal and vertical kernels $\boldsymbol{K}_h$ and $\boldsymbol{K}_v$. These can be constructed by repeated convolution, with intermediate results employed to compute the noise variance at each successive level of the subband pyramid.

2. Convolve the noise ACF with each of these kernels separably, treating entries outside the kernel as zero, and returning only the central portion of each convolution. This will produce a 2D array equal in size to the corresponding kernel in each dimension.

3. Multiply this convolution product element-wise with each of $\boldsymbol{K}_h$ and $\boldsymbol{K}_v$ separably. In other words, multiply each row by the corresponding element of $\boldsymbol{K}_v$ and each columns by the corresponding element of $\boldsymbol{K}_h$.

4. The sum of all elements of the resulting matrix is then the noise variance for the subband in question.

Actually, the above algorithm could be redesigned, further exploiting the separability of the kernel involved, so that no fully two dimensional matrix is ever constructed. This is probably not worth the effort unless memory is extremely scarce. Also, it should be clear how this algorithm generalizes to non-separable kernels.

This algorithm has all the expected properties. For example, if the noise is white, then the ACF has only one non-zero element and the convolution becomes scalar multiplication by the variance of the underlying grain noise. Subsequently, the computed noise variance is simply equal to the grain variance times the sum-of-squares of the kernel coefficients, which is the correct result for the variance of a weighted sum of identical Gaussian variables.

In summary, the noise distribution $p_n$ for a single subband coefficient is modeled as the product of three independent univariate Gaussian distributions, one for each colour channel. The variance of each of these distributions may be efficiently computed from the noise ACF and the subband kernels employed, as described above.

### 10.2.2 Coefficient Distribution

The development of a subband coefficient distribution $p_x$ is somewhat difficult. Generally, some model which can well approximate the expected range of input distributions is sought, but this is not enough. It must additionally be possible to produce a reliable estimate of this distribution given only an observed distribution corrupted by Gaussian noise of (approximately) known variance.

As shown in section 8.4, under the usual additive model the addition of noise corresponds to convolution of the image PDF with the noise PDF. Thus if the observed noisy PDF can be obtained and the noise distribution is known, it is theoretically possible to "deconvolve" the observed distribution, presumably through application of the inverse filter, to recover the clean PDF. However, such inverse filtering corresponds to a sharpening operation which is extremely sensitive to noise and other inaccuracies in the available data. As studied extensively elsewhere (e.g. [4]) inverse filtering is a very ill-conditioned operation and therefore this direct approach is certainly very difficult and possibly entirely infeasible, especially in three dimensions.

Instead, parameterized distribution models can be employed. Parameterized models alleviate the ill-conditioned nature of the distribution recovery problem by restricting the allowable models to a finite-dimensional set. In other words, only a small number of variables need to be recovered so the construction of robust estimators is easier. Further, the parameterized model can be designed with the distribution recovery task in hand.

In the work of Simoncelli and Adelson on monochrome Bayesian coring [56], where only a univariate model is required, a generalized exponential distribution of the form

$$p_x(x) \propto \exp(-|x/s|^p) \tag{10.15}$$

was employed.[1] The scale parameter $s$ controls the distribution width, while the exponent $p$ determines the weight contained in the tails of the function. If $p = 2$ then a Gaussian distribution is achieved while if $p = 1$ then the model is Laplacian (bi-exponential). Other values give intermediate or more extreme shapes. This model was chosen by examining the general shape of subband coefficient distributions, and was found to be a reasonably accurate

[1] The normalization constant of $p/(2s\Gamma(\frac{1}{p}))$ has been omitted here for simplicity.

approximation, with the shape parameter $p$ usually in the range of 0.5 to 1. Importantly, the second and fourth moments of such a distribution corrupted by white Gaussian noise are analytically available. With the aid of a symbolic math package these can be shown to be:

$$m_2 = \sigma_n^2 + \frac{s^2\Gamma(\frac{3}{p})}{\Gamma(\frac{1}{p})} \tag{10.16}$$

$$m_4 = 3\sigma_n^4 + \frac{6\sigma_n^2 s^2\Gamma(\frac{3}{p})}{\Gamma(\frac{1}{p})} + \frac{s^4\Gamma(\frac{5}{p})}{\Gamma(\frac{1}{p})} \tag{10.17}$$

where $\sigma_n^2$ is the variance of the corrupting noise, and $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,dt$ is the well-known gamma function. Thus, knowledge of the noise variance and the second and fourth moments of the observed noisy distribution is sufficient to estimate the parameters $s$ and $p$.

Exactly how this inverse computation is to be done is not explicitly detailed in [56]. One possibility, tried initially, is multi-variable minimization over the parameters $s, p$, using an error function which penalizes the difference between the observed moments and those which would occur at any given set of parameter values. This works, but it is actually quite expensive and not very robust, because the solution lies within an extremely narrow trough in the resulting error surface, making numerical minimization difficult and very sensitive to the initial guess. A much better approach is to note that, by rearranging equation 10.16, $s$ is completely determined terms of $p$, $m_2$, and $\sigma_n^2$ as

$$s = \sqrt{\frac{(m_2 - \sigma_n^2)\Gamma(\frac{1}{p})}{\Gamma(\frac{3}{p})}}. \tag{10.18}$$

This allows univariate minimization of $p$ alone, which is numerically much easier and faster.

In this spirit, a three-dimensional distribution model which depends on only a few easily recoverable parameters is sought. An obvious first step is the examination of the shape of actual (clean) subband coefficient distributions. From the experiments of chapter 7 we expect such distributions to show strong correlations between the channels, and figure 10.5 confirms this. This figure shows the 3D distribution of the $H_1$ (highest-frequency horizontal) subband of the Monarch test image shown in figure 6.1, which is quite representative of the distributions from various test images and subbands. As first observed in section 7, image features tend to have very similar magnitudes in each channel, a phenomenon initially seen using the much simpler

technique of plotting the difference between adjacent pixels, in figure 7.1. The simple form of this distribution is encouraging and suggests that a simple parametric model can indeed be found.

It is known that each of the marginal distributions is approximately Laplacian, i.e. $p \approx 1$ in equation 10.15, so the overall PDF certainly cannot be a multi-variate Gaussian distribution. Some other parametric model must be found. One possibility is a simple generalization of 10.15 to the multi-variable case:

$$p_x(\boldsymbol{x}) \propto \exp(-|\boldsymbol{A}\boldsymbol{x}|^p) \tag{10.19}$$

were $\boldsymbol{A}$ is a $3 \times 3$ transformation matrix which defines the orientation and scale of the distribution. This model is elegant and probably quite adequate, and in fact noise reduction has successfully been carried out with a function of this form and hand picked parameters $\boldsymbol{A}$ and $p$. However, it suffers from a serious drawback: there is no easy way to estimate the required parameters from noisy data. The estimation procedure used in the univariate case does not generalize. First, there are 15 different possible fourth moments of a three-variable distribution, and it is not at all obvious how to use these values to estimate the shape parameter $p$. For that matter, there is no analytic expression for the moments of this type of distribution corrupted by multi-variate Gaussian noise, except in special cases. That is, the resulting integrals are in general not expressible in terms of the Gamma function, nor does it seem likely that they can be evaluated in terms of other simple integrals. Of course, numerical solutions are possible but these are expensive to compute, and in any case it is not clear how $p$ may be estimated even if these moments could be evaluated.

It is true of course that the full brunt of statistical estimation theory could be applied to solve these problems, but given that the choice of model was arbitrary to start with, this seems to be unnecessary work. Instead, we are free to pick a different model which is more easily estimated.

First, in the spirit of parametric modeling, it is useful to consider whether all of the degrees of freedom afforded by the matrix $\boldsymbol{A}$ of equation 10.19 are truly necessary. In particular, this matrix can represent arbitrary orientations, yet we know (or assume) that the colour channels

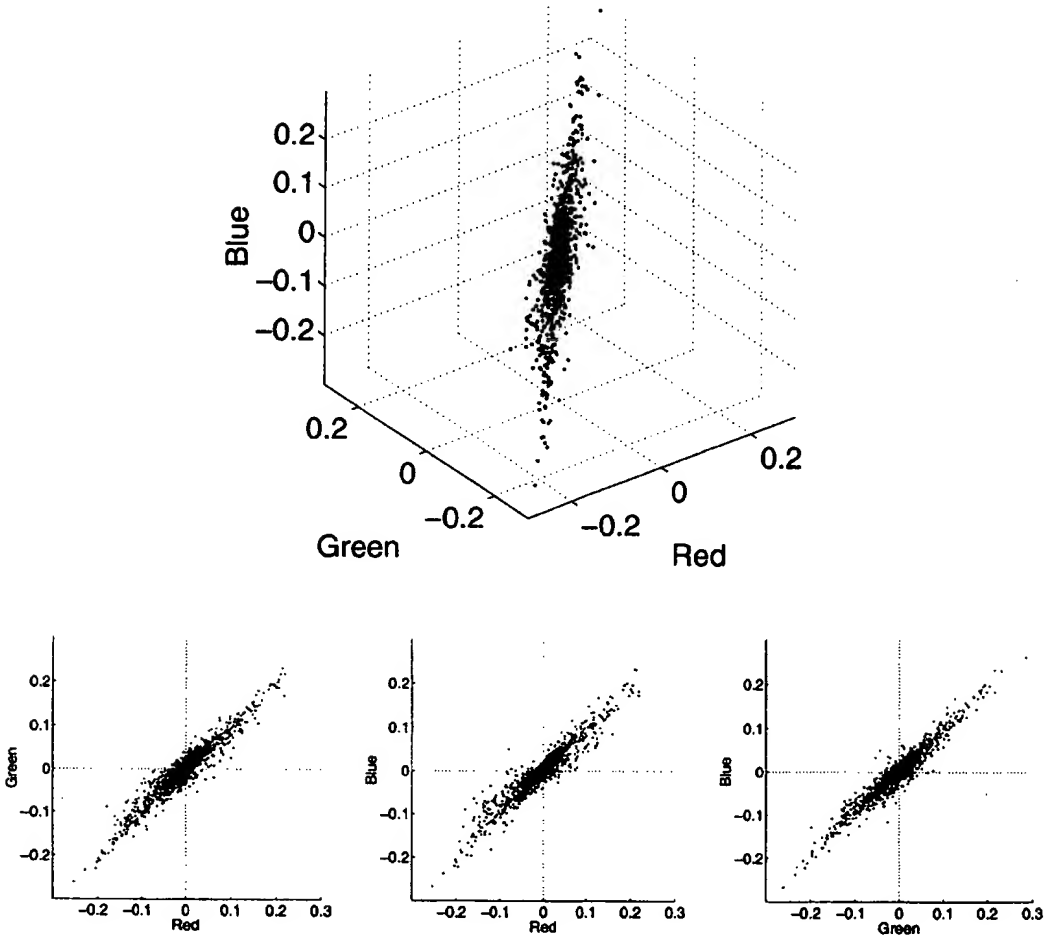



Figure 10.5: Study of simultaneous spectral-spatial correlation: the probability distribution of subband coefficients taken from the noiseless "Monarch" test image.

will always be strongly correlated, causing the subband distribution to always point along the same major axis. Thus it seems that two rotational degrees of freedom are unneeded. These can be eliminated by defining the model in a coordinate system where one axis is aligned with the major axis of RGB colour-space. Equivalently, for the purposes of evaluating the coefficient distribution model, we can work in a colour-space where one coordinate is proportional to $R+G+B$, a quantity related to the perceived brightness of a colour and which will therefore be referred to as "intensity" and denoted by $I$. Note that if we allow later transformation of the remaining two coordinates, any orthonormal transform which defines $I \propto R+G+B$ is adequate, as all such transforms differ only by a linear transformation in the other two coordinates. The commonly used YIQ and YUV colour spaces do *not* have this property, as their "luminance" $Y$ is an unequally weighted combination of $R, G, B$ designed to approximate the human visual system's spectral response. One transformation which *does* have the desired property is

$$M = \frac{1}{6}\begin{bmatrix} 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \\ -2\sqrt{3} & \sqrt{3}+3 & \sqrt{3}-3 \\ -2\sqrt{3} & \sqrt{3}-3 & \sqrt{3}+3 \end{bmatrix}. \tag{10.20}$$

We may then write

$$\begin{bmatrix} I \\ S \\ T \end{bmatrix} = M \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{10.21}$$

Here the remaining two coordinates have been arbitrarily named $S$ and $T$. These two values correspond to the "chromaticity" of the colour represented by $R, G, B$ and will both be zero for perfect grays, that is, colours where $R = G = B$. It is clear that for subband coefficients, $S$ and $T$ will vary much less than the intensity coordinate $I$. In other words the subband signals are predominantly "gray", as demonstrated in chapter 7. This transformation is orthonormal, as may be easily verified.

The ability to control the shape of the distribution through a parameter such as the exponent $p$ seems to be a desirable feature for a distribution model, but as shown above this exponent cannot be evaluated when the problem is multi-variate. In contrast, this exponent can be estimated for a univariate distribution fairly easily. Therefore the generalized exponential distribution of equation 10.15 could be used as one factor of a separable function. Since

the intensity axis has unique importance, we might try a model of the form

$$p_x(I,S,T) \propto \exp(-|I/s|^{p_1}) \exp(-|\boldsymbol{B}\left[\frac{S}{T}\right]|^{p_2}). \tag{10.22}$$

Note that this model is defined in terms of $I,S,T$ coordinates, thus it is immediately aligned to the correct orientation. The intensity coordinate $I$ is considered to have its own generalized exponential marginal distribution, with some scale $s$ and exponent $p_1$ which are estimated by the previously discussed univariate moment matching technique. The chromaticity coordinates are also allowed their own generalized distribution in two variables, defined by transformation matrix $\boldsymbol{B}$ and exponent $p_2$. Here we run into the same problem again, as the integrals necessary for computing the moments of this distribution are not analytically solvable, in general.

However, if we set $p_2 = 2$, corresponding to a multi-variate Gaussian distribution of $S$ and $T$, then the moments can be computed because a Gaussian distribution is separable (in the appropriate coordinate system). In fact there is no need to perform any numerical parameter estimation at all, as the parameter estimation task simply involves the construction of a multi-variate Gaussian distribution which fits the observed population, a well studied problem. The matrix $B$ takes on a particularly simple form, becoming equal to square root of twice the covariance matrix between $S$ and $T$. Letting

$$\boldsymbol{c} = \left[\frac{S}{T}\right]. \tag{10.23}$$

the required covariance matrix can be denoted $\boldsymbol{R_c}$ and could be called the "chromatic" covariance matrix. This matrix can be simply estimated from noisy observations, as will be shown shortly.

The resulting model is then

$$p_x(I,\boldsymbol{c}) \propto \exp(-|I/s|^p)\ \exp\left(-\frac{\boldsymbol{c}^T\boldsymbol{R_c}\boldsymbol{c}}{2}\right). \tag{10.24}$$

This equation might be objected to for many reasons. For example, there is no reason to suppose that the true chromatic distribution will be roughly Gaussian ($p_2 = 2$) as opposed to, say, roughly Laplacian ($p_2 = 1$). More fundamentally, why should the model be separable in intensity and chromaticity? These and other objections turn out to be unimportant, but this is not clear without the benefit of further analysis and experimentation, to be detailed in

the next section. While it is true that equation 10.24 has many shortcomings, it turns out that effective multi-channel noise reduction depends on only a few important features of the distribution model. These include the strong correlation between channels (implicit in the use of the intensity coordinate $I$) and the narrowness of the chromatic distribution as opposed to the intensity distribution (which is captured by the above model regardless of the exact distribution shape).

Note also that the use of a distribution which is defined separably in a transformed colour space does imply that noise reduction is done independently between channels in a transformed colour space. Far from it. First, the colour space in which Bayesian estimation is performed is irrelevant, as colour-space transformations just correspond to global rotations of the various distributions involved. Since application of a colour-space transformation to an observed image rotates the noise along with the signal, the final outcome is unchanged. Second, even though the coefficient distribution is modeled separably, it is separable in a transformed set of axes and not in RGB space. Conversely, the noise is modeled as independent between channels, hence it is separable in RGB space. There is no coordinate system in which *both* these distributions are simultaneously separable, thus the estimation integral of equation 10.8 remains fully three-dimensional.

It remains to discuss the precise procedure for estimating the various parameters of equation 10.24. First, the input data, being the set of all multi-channel coefficients for the particular subband being modeled, must be transformed into $I, S, T$ space by multiplication by the matrix $M$ of equation 10.20. Before any parameter estimation can be done, the noise model must also be transformed into this coordinate system. As discussed in the previous section, the noise is modeled as a multi-variate Gaussian distribution aligned to the $R, G, B$, axes. Applying the usual formula for coordinate transformation of variances gives

$$\begin{bmatrix} \sigma_I^2 & \sigma_{IS} & \sigma_{IT} \\ \sigma_{SI} & \sigma_S^2 & \sigma_{ST} \\ \sigma_{TI} & \sigma_{TS} & \sigma_T^2 \end{bmatrix} = M \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_G^2 & 0 \\ 0 & 0 & \sigma_B^2 \end{bmatrix} M^T \tag{10.25}$$

where $\sigma_R^2$, $\sigma_G^2$, and $\sigma_B^2$ are the computed noise variances of the red, green, and blue channels respectively. Note that the resulting noise covariance is not diagonal, indicating that the noise

distribution is not aligned to the new coordinate system.

Then, the parameters $s$ and $p$ can be computed by applying the univariate distribution modeling technique discussed earlier to the marginal distribution of $I$ alone. This procedure requires the variance of the noise corrupting the observed values of $I$, which is just the value $\sigma_I^2$ from equation 10.25.

Finally, the joint distribution of $S$ and $T$ must be described using a multi-variate Gaussian model, defined by the chromatic covariance matrix $\boldsymbol{R_c}$. If $S$ and $T$ were not noise corrupted, then $\boldsymbol{R_c}$ could be estimated very easily by the usual technique of taking the average across all samples of the products $S^2$, $T^2$, and $ST$. However $\boldsymbol{R_c}$ must be estimated in the presence of noise, so a more sophisticated approach is needed. Let

$$\boldsymbol{d} = \boldsymbol{c} + \boldsymbol{n} \tag{10.26}$$

where $\boldsymbol{d}$ represents the observed noisy values of $\boldsymbol{c}$ and $\boldsymbol{n}$ is the corrupting noise. Then,

$$\mathrm{E}\left\{\boldsymbol{d}\boldsymbol{d}^T\right\} = \mathrm{E}\left\{\boldsymbol{c}\boldsymbol{c}^T\right\} + \mathrm{E}\left\{\boldsymbol{n}\boldsymbol{n}^T\right\} + \mathrm{E}\left\{\boldsymbol{c}\boldsymbol{n}^T\right\} + \mathrm{E}\left\{\boldsymbol{n}\boldsymbol{c}^T\right\}. \tag{10.27}$$

But, by assumption, the noise is independent of the signal, so the cross-terms vanish, giving the simple formula

$$\boldsymbol{R_c} = \boldsymbol{R_d} - \boldsymbol{R_n}. \tag{10.28}$$

This formula is directly applicable because $\boldsymbol{R_d}$ is the covariance matrix of the observed values, and the noise covariance matrix $\boldsymbol{R_n}$ is equal to

$$\boldsymbol{R_n} = \begin{bmatrix} \sigma_S^2 & \sigma_{ST} \\ \sigma_{TS} & \sigma_T^2 \end{bmatrix} \tag{10.29}$$

with all elements defined by equation 10.25. In principle, $\boldsymbol{R_c}$ is positive-definite as a covariance matrix must be, but there is no reason this must hold in application as the above technique merely generates an *estimate* of the correct matrix. In practice, the estimate of $\boldsymbol{R_c}$ derived according to equation 10.28 may be have negative eigenvalues, which would make it unusable in equation 10.24. For example this could occur if the noise variance is over-estimated. Noting

that the observed covariance matrix $\boldsymbol{R_d}$ is will always be positive definite, this suggests the more robust estimator

$$\boldsymbol{R_c} = \boldsymbol{R_d} - \alpha \boldsymbol{R_n}. \tag{10.30}$$

where the coefficient $\alpha$ is initially set equal to one, and is gradually reduced until $\boldsymbol{R_c}$ becomes positive definite. There is no theoretical justification for this approach, but it does compensate for over-estimates of the noise variance and seems to work well in practice. One difficulty is that when a value of $\alpha$ is found that just makes $\boldsymbol{R_c}$ positive definite, the resulting estimate will have an eigenvalue very close to zero and will thus be near-singular. Clearly more research is needed in this area, yet this simple approach is usually sufficient.

With the parameters $s$, $p$ and $\boldsymbol{R_c}$ thus defined, equation 10.24 can be evaluated directly. Note however that this equation is defined in the $I, S, T$ coordinate system so RGB input parameters must be transformed before the model can be evaluated.

### 10.2.3 Efficient Implementation

With the appropriate noise model and prior coefficient distribution, equation 10.8 can be applied directly. The required integrals are again of a form which is difficult and perhaps impossible to evaluate analytically. We are forced to use a numerical solution.

There are many possible numerical integration techniques which could be employed, but it is clear that direct evaluation of equation 10.8 for each observed RGB triplet of coefficients would be far to slow to be practical for real images which have millions of pixels. The solution is to note that, for any given noise distribution $p_n$ and coefficient distribution $p_x$, the estimate $\tilde{\boldsymbol{x}}$ is a function only of the observed value $\boldsymbol{y}$, both of these quantities being RGB triplets of subband coefficients. Therefore it is possible to precompute a lookup table which gives the estimated value on a grid of possible coefficient triplets. This table maps triplets to triplets and is therefore a 3D table of vector values, or equivalently three parallel 3D tables of scalars, one each for red, green, and blue. Since the coring function represented by this table is expected to be smooth (and it is certainly continuous at the very least) intermediate values can be computed by linear interpolation.

Examining equation 10.8 again, it is a quotient of two integrals:

$$\tilde{x} = \frac{\int x p_x(x) p_n(y-x)\, dx}{\int p_x(x) p_n(y-x)\, dx} \tag{10.31}$$

There are again many possible ways to (numerically) compute the required integrals. Perhaps the simplest possible approach involves evaluating the functions $x$, $p_x(x)$, and $p_n(y-x)$ on a regular grid. Then, the integral can be approximated taking the appropriate point-wise products and summing. This is a very simple quadrature rule and no doubt much more efficient integration techniques could be applied, e.g. error-bounded adaptive approaches.

However, when 10.8 must be evaluated at every point on a regular grid to construct a lookup table, this simple integration technique lends itself perfectly to a very efficient algorithm. The key insight is that the required integrals are actually *convolutions* of some function with $p_n$. Specifically, the denominator is

$$\int p_x(x) p_n(y-x)\, dx = [p_x \star p_n](y) \tag{10.32}$$

and the numerator is

$$\int x\, p_x(x)\, p_n(y-x)\, dx = [(x p_x) \star p_n](y). \tag{10.33}$$

Evaluation of the estimation equation for any particular value of $y$ corresponds to computing the value of the above convolutions at a *single point*, and integration by summing over regularly spaced samples is just the usual discrete convolution algorithm. Correspondingly, these convolutions may be computed at all points on a regular grid simply by shifting the pre-computed values of $p_n$ relative to the other functions. That is, after the functions $x$, $p_x$, and $p_n$ are evaluated *once* at each grid point, construction of a regularly spaced table of values can be done by three-dimensional discrete convolution with $p_n$. Even better, the noise model $p_n$ is *separable* because the noise is assumed independent between channels. Therefore, this 3D convolution can be implemented very efficiently by performing successive 1D convolutions in each dimension.

As always with discrete convolution, edge conditions need to be defined. It has been found that in practice the simple approach assuming zero outside the tabulated range of each function is perfectly adequate, as long as the table has sufficient range that the various probability

densities involved are close to zero at the edges of the table. This is also efficient, because it results in many zero values being encountered during convolution.

After both the numerator and denominator are computed on regular grids, the final estimation table is generated by point-wise division. This division operation makes it unnecessary to normalize the distribution models; in fact it is advantageous to scale the distributions by a large constant factor so that accuracy is maintained during the convolution operation. Also, it is important that one of the grid points be precisely at the origin, where the optimal estimate will always be zero (due to the even symmetry of the models involved). Otherwise, linear interpolation between table elements will produce unduly large outputs near the origin, and noise reduction will suffer.

The questions of grid range and resolution are of course important. Obviously we'd like the grids to cover the entire range of possible observed values, but most values will be very near zero, so this requirement conflicts with the desire for as much resolution as possible, because only a finite amount of memory is available. At least one (scalar) table for each channel is required, and if many frames with the same image and noise characteristics are to be processed (as in film footage) then ideally one set of tables for each subband should be stored. If these tables contain single precision floating-point values, a three level three direction subband transform is employed, and the tables are computed at a resolution of $50^3$, then 13,500,000 bytes of storage are needed, not an insignificant amount. Further, this amount scales cubicly with the resolution or range. Thus both of these quantities must be kept to a minimum. Fortunately, the estimation integral appears to go asymptotically to a linear function as $|y|$ increases, so linear extrapolation of table values is a reasonable technique to handle out-of-range inputs. With this approach, a range of plus or minus five standard deviations of the input distribution and a resolution of 50 grid points has been found to be more than sufficient.

## 10.3 Results

The final multi-channel noise reduction algorithm, which might be called Multi-Channel Bayesian Subband Coring (MCBSC), is:

1. For each channel, generate a parametric noise model using the techniques outlined in chapter 4.

2. Decompose the image into subbands using a pyramidal transform such as the one designed in section 10.1.1.

3. For each subband:

   - Based on the noise models, compute the noise variance in the subband as described in section 10.2.1.
   - Estimate the noiseless subband coefficient distribution as described in section 10.2.2.
   - Generate a lookup table for each colour channel, using the convolution-based technique of section 10.2.3.
   - Core the subband coefficients by looking them up in these multi-channel tables, using linear interpolation between grid points and linear extrapolation off the edges of the table.

4. Invert the subband transform to produce the cleaned image.

To show that this algorithm effectively exploits multi-channel correlations, a simple test case is shown in figure 10.6. The colour "monarch" image from section 6.1 is shown. This image is then corrupted with white Gaussian nose which is independent between channels, having a standard deviation of 30 as compared the 255 levels available in the eight-bit representation used.

Figure 10.7a shows this image cleaned by single-channel Bayesian coring. Each channel is decomposed using the subband transform developed in section 10.1.1. The resulting subbands are processed independently using the 1D Bayesian coring technique from [56], which uses the univariate generalized exponential distribution as a coefficient model, as described in section 10.2.2. Substantial noise reduction is achieved, but image details are somewhat blurred.

Finally, in figure 10.6b the new MCBSC algorithm described in this chapter used. Slightly more noise is removed as compared to the single channel algorithm, but more importantly this image is noticeably sharper than in the single channel case. In fact, fine details (e.g. the

antennae, colour divisions in the wings) are almost as sharp as the original noiseless image of figure 10.6a. Also, the residual noise is less saturated (it tends to be gray rather than coloured) which decreases its visibility.

A test on real film grain is of course required. Figure 10.8a presents the context surrounding the usual "eye" test image. Note the fine detail in the hair. Figure 10.8b shows this image cleaned by MCBSC. The fine detail is almost entirely preserved, and the image appears sharp, yet there is significant reduction in grain. This test was performed in the density domain, using noise models obtained through the usual techniques of chapter 4.

These simple comparisons serve to validate the basic algorithm and concept. More formal testing will be conducted in the next section.

### 10.3.1 Quantitative Comparison

This section presents quantitative experiments which demonstrate the effectiveness of the new algorithm. A number of difficulties are immediately encountered in trying to design such an experiment. Chief among these is the lack of a suitable quality metric for images. The venerable mean-square-error (MSE) metric has long been used to measure algorithm performance, along with derivative measures such normalized MSE (NMSE) and Signal-Noise-Ratio in decibels (SNR). However all metrics of this type suffer from serious problems. These all stem from the fact that MSE considers only differences at each pixel, while more sophisticated image features are very important in terms of subjective image quality. In particular MSE is not very sensitive to qualities like sharpness, preservation of small details, and colour shifts.

This problem has led to the suggestion of a number of alternative measures. For example, computation of MSE in a perceptually uniform colour space, such as the CIE $L^*u^*v^*$ system, makes this metric more sensitive to chromatic effects, and the resulting estimator is known as the Normalized Colour Difference (NCD) [17].

Fundamentally however point metrics are just not capable of discerning many significant image effects, for much the same reasons that it is not possible to perform effective noise reduction by examining only single pixels in isolation. Structure is very important in images, and structure by definition spans multiple pixels. The image processing community lacks a

robust error metric which is based on image features rather than pixel values, though there is research in this direction. In the spirit of the algorithms developed in this work, perhaps point metrics applied to subband coefficients would be effective.

Still, the MSE based metric of Signal-to-Noise Ratio in decibels ($SNR_{dB}$) was eventually chosen for the current experiments. Despite its many shortcomings, it does provide an effective measure of gross noise reduction performance, and it is certainly easy to calculate. More importantly, reporting the results in this format allows comparison with other reported noise reduction results. This metric is defined as

$$SNR_{dB} = 10 \log_{10} \left( \frac{\sum x_i^2}{\sum (\tilde{x}_i - x_i)^2} \right) \tag{10.34}$$

where $x_i$ is an element of the original noiseless image, $\tilde{x}_i$ is an element of the cleaned estimate, and the summations are taken over all channels of all pixels. This variant of MSE was chosen as it is easier to interpret than MSE (which is relative to signal characteristics) or NMSE (which produces unintuitive small fractions.) However the chosen metric may be easily converted into other MSE-based forms if desired. Of particular interest is Signal-to-Noise Ratio Improvement (SNRi) which may be computed by subtracting the SNR of the unprocessed noisy image.

Of course, equation 10.34 requires that clean images are available, which is not possible in the case of film grain. Instead, film grain must be simulated. This was done by generating Gaussian white noise independently in each channel, then convolving the noise field with a narrow 2D filter kernel (three pixels wide and separable, pseudo-Gaussian with coefficients [0.11 0.78 0.11]) which simulated the effects of optical degradation in the scanning process. The resulting non-white noise field was added to the original clean images. Note that the dependence of noise magnitude on signal intensity was not simulated, and issues relating to this will be discussed later.

In the final experiment, two colour test images were used, "Monarch" and "Lena". these images were corrupted by noise at five different levels, measured in terms of the ratio of the noise standard deviation to the maximum representable pixel value (the source images were encoded with eight bits per colour channel, so this maximum value is 255.) Technically this noise metric describes a type of Signal-to-Noise ratio but so as not to cause confusion with $SNR_{dB}$, it will

be referred to simply as "noise level". For completeness, both white and non-white noise was tested. The $SNR_{dB}$ is reported for each image and noise level before any processing is applied and after treatment by different noise reduction algorithms.

Obviously the noise reduction algorithms to be tested must include both single and multi-channel Bayesian subband coring. However other algorithms must be tested before meaningful comparisons can be made. The additional techniques eventually chosen were

- median filtering, because it is in some sense a widely used "baseline" technique,
- the adaptive Wiener filter, chosen as being representative of more sophisticated pixel-by-pixel filters, and
- block-wise spectral power subtraction, which was the most effective block-based technique encountered in chapter 5.

Other techniques were excluded for various reasons. Convolution-based approaches, including the monochrome Wiener filter, are easily surpassed even by relatively simple techniques such as the median filter because of the blurring they cause. The colour Wiener filter is ineffective without *a-priori* knowledge of the cross-power spectra of the clean image, so it is not a practical technique as discussed in section 6.2.2. Finally, the various Vector Median filters of section 6.4.2 did not display significantly better performance on film grain than the simple median filter and were felt to be not worth comparing unless the usual median filter turned out to be promising.

Unlike earlier experiments, all images were processed in the intensity domain, however this is misleading because the (synthesized) noise is Gaussian in this domain, which is the point of processing in the density domain.

Note that all algorithms were run completely "blind" in the sense that they did not have any *a-priori* information about the clean image or noise characteristics. Whenever noise PSD estimates or power levels were needed, these were extracted directly from the noisy image using the techniques presented in chapter 4. This is an important point as some previous multi-channel noise reduction studies (e.g. [21]) provide certain information during testing which would not be available in practice. It also serves to highlight the advantages of those algorithms

which use little or no prior information (e.g. median filter) and are thus not susceptible to noise statistics estimation errors.

The results of these experiments are summarized in tables 10.3.1 and 10.3.1. As expected, the MCBSC algorithm generally out-performed all others. In fact, it produced the maximum $SNR_{dB}$ in every case except for being marginally inferior to the adaptive Wiener filter in three of the lowest noise cases. This was followed by the single channel Bayesian coring algorithm, which usually out performed the remaining approaches. In rough order of decreasing $SNR_{dB}$ the remaining algorithms were the adaptive Wiener filter, the median filter, and spectral subtraction. In general as the noise level increased, the multi-channel algorithm displayed increasingly better performance relative to the other approaches. Note that film is quite noisy, so the 1/5 maximum value test case best approximates grain noise. At this level, MCBSC out-performs all other approaches by at least a few decibels.

What is not shown in these tables is the quality and appearance of the resulting images. Generally, the multi-channel Bayesian estimates looked quite good, sharp with essentially no visible artifacts, although faint ringing was sometimes visible around very sharp edges and small features. The remaining noise was achromatic, which helped to hide it. The single channel Bayesian estimates were also acceptable, though somewhat less sharp, and the remaining noise was somewhat colourful. The median filter, although acceptable in terms of the chosen error metric, was completely unacceptable visually, especially at high noise levels, due to the mottling and quantization artifacts. The adaptive Wiener filter was satisfactory at low noise levels, but rapidly succumbed to "speckle" artifacts brought on by its finite window size. Finally, although the results obtained with Spectral subtraction appeared fairly good visually, this algorithm performed surprisingly poorly in terms of the quality metric. This is probably due to the faint block structure visible in the resulting image, as well as the subtle ringing which sometimes occurred around edges.

### 10.3.2 Discussion

Based on the theoretical analysis of the previous three chapters, this chapter has detailed the construction of a new algorithm designed to take advantage of the extensive cross-channel

Monarch image, white noise

| Noise level | Noisy | Multi | Single | Med | AWF | SPS |
|---|---|---|---|---|---|---|
| 1/30 | 23.411 | 26.167 | 26.063 | 23.584 | 27.362 | 24.551 |
| 1/20 | 19.928 | 25.193 | 23.849 | 22.415 | 24.531 | 22.126 |
| 1/10 | 13.887 | 22.31 | 20.092 | 19.168 | 19.51 | 17.241 |
| 1/5 | 7.8901 | 18.837 | 16.256 | 14.64 | 14.163 | 11.713 |
| 1/2 | -0.072 | 13.833 | 10.741 | 7.4561 | 6.5962 | 4.0017 |

Monarch image, non-white noise

| Noise level | Noisy | Multi | Single | Med | AWF | SPS |
|---|---|---|---|---|---|---|
| 1/30 | 27.266 | 26.979 | 28.123 | 24.149 | 29.636 | 26.998 |
| 1/20 | 23.784 | 25.778 | 25.777 | 23.305 | 26.86 | 24.628 |
| 1/10 | 17.731 | 23.787 | 21.554 | 20.627 | 21.746 | 20.084 |
| 1/5 | 11.747 | 19.802 | 17.591 | 16.391 | 16.405 | 14.94 |
| 1/2 | 3.7762 | 14.271 | 12.436 | 9.3389 | 8.9626 | 7.5535 |

Table 10.2: Comparison of noise reduction algorithms on the Monarch image. All results reported in $SNR_{dB}$ (higher numbers better). Noisy = $SNR_{dB}$ before any processing, Multi = multi channel Bayesian coring, Single = single channel Bayesian coring, Median = 3x3 median filter, AWF = adaptive Wiener filter on 3x3 window, SPS = spectral power subtraction on half-overlapped 24x24 blocks.

Lena image, white noise

| Noise level | Noisy | Multi | Single | Med | AWF | SPS |
|---|---|---|---|---|---|---|
| 1/30 | 24.408 | 29.125 | 28.027 | 25.592 | 28.274 | 26.83 |
| 1/20 | 20.879 | 27.158 | 25.632 | 24.3 | 25.423 | 23.729 |
| 1/10 | 14.844 | 23.929 | 21.886 | 20.757 | 20.389 | 18.264 |
| 1/5 | 8.851 | 20.067 | 18.229 | 15.902 | 15.115 | 12.698 |
| 1/2 | 0.8928 | 14.748 | 11.79 | 8.5161 | 7.4052 | 4.8375 |

Lena image, non-white noise

| Noise level | Noisy | Multi | Single | Med | AWF | SPS |
|---|---|---|---|---|---|---|
| 1/30 | 28.255 | 30.785 | 30.273 | 26.223 | 30.751 | 29.749 |
| 1/20 | 24.73 | 28.577 | 27.65 | 25.255 | 27.805 | 26.649 |
| 1/10 | 18.69 | 25.071 | 23.391 | 22.236 | 22.629 | 21.345 |
| 1/5 | 12.698 | 21.242 | 19.577 | 17.699 | 17.366 | 15.992 |
| 1/2 | 4.7496 | 16.366 | 14.074 | 10.44 | 9.8974 | 8.4846 |

Table 10.3: Comparison of noise reduction algorithms on the Lena image. All results reported in $SNR_{dB}$ (higher numbers better). Noisy = $SNR_{dB}$ before any processing, Multi = multi channel Bayesian coring, Single = single channel Bayesian coring, Median = 3x3 median filter, AWF = adaptive Wiener filter on 3x3 window, SPS = spectral power subtraction on half-overlapped 24x24 blocks.

correlation found in real colour images. The MCBSC algorithm has been demonstrated to out perform a variety of other noise reduction techniques, both in terms of error metrics and image quality. It also largely achieves the desiderata of section 1.3:

- The technique introduces few visual artifacts. In particular it retains sharpness and fine detail very well.
- The algorithm is stable because the output is a continuous function of the input, as discussed below.
- A great deal of noise is nonetheless removed.
- Coupled with automatic noise estimates, the whole process can be performed reliably without user intervention.
- The same noise estimates and image models can be used for all frames of a sequence. Therefore, after a certain amount of precomputation, the algorithm is very fast because it involves nothing more than separable convolution and linear interpolation of lookup tables.

The algorithm also has a number of other useful properties:

- If applied to a noiseless image MSBSC produces no visible change, while at the same time it performs increasingly well relative to other techniques as noise power increases.
- Viewed as a function which produces a cleaned estimate from a noisy observation, $\tilde{\boldsymbol{x}} = MCBSC(\boldsymbol{y})$, the algorithm function is continuous and smooth. This is critical for noise reduction, because it prevents small changes in input value from producing large changes in the output. Without this condition, the algorithm could not be temporally stable, for example.
- Similarly, while of course highly non-linear, the $MCBSC$ function is invariant under affine transformations, i.e. $aMCBSC(\boldsymbol{y}) + c = MCBSC(a\boldsymbol{y} + c)$. This is important because it means the algorithm is not sensitive to overall brightness and contrast levels.

- Although this was not developed due to the emphasis on film grain, the noise to be removed need not be independent between channels as long as it follows a multi-variate Gaussian distribution. In this case a colour-space transform may be applied before processing to decorrelate the noise between channels. (Of course this same transform would also need to be applied to the coefficient model of equation 10.24.) In this fashion the noise distribution remains separable and channel correlated noise may be processed with almost no loss of efficiency.

Perhaps most importantly, MCBSC seems to be extremely robust to errors in the noise and image distribution estimates on which Bayesian estimation so crucially depends. Initial experiments show that significantly perturbing the distributions involved has little effect on the quality of the resulting image. Examples of perturbations include changing the exponent $p$ and chromatic covariance matrix $\boldsymbol{R}_c$ of equation 10.24, changing the noise PSD models and auto-correlation functions, and perturbing the computed noise variances for each subband. Perturbations of these quantities produced almost imperceptible changes in the output image up to perhaps 50% error, depending on the value disturbed. This property is very important because all of these quantities must be estimated from observed noisy data, and are thus subject to error. In a sense, it seems that this algorithm depends almost exclusively on the general shape of the distributions rather than the exact details.

However, all is not perfect. First of all, the dependence of film grain noise variance on signal magnitude discussed in chapter 3 was not modeled or exploited. This was done to reduce the complexity of the resulting algorithm, and may or may not be an important issue. In principle, this could be handled by adjusting the noise model dynamically for different regions of the image. In practice however this problem does not seem very serious, in that there is little difference in appearance between dark regions (where the noise is greatest) and light regions (where the noise is minimal) in processed images.

Also, the dependence of MCBSC, or indeed any multi-channel algorithm, on correlations between colour channels is a weakness as well as a strength. In particular, regions of highly saturated red, green, or blue detail are incorrectly smoothed, because such features appear in only one channel. There are various *ad hoc* approaches which might alleviate this problem,

such as reducing the amount of noise reduction in large saturated areas, but no good theory to address this problem.

Further, the subband transform developed in this chapter is rather arbitrary. This is perhaps the weakest design choice in the current algorithm. Based on the experience of the author, it is thought that a more frequency selective transform – having narrower subbands in the high frequency region – could result in better noise reduction. However, there is a fundamental lack of theory in this area, and little previous work investigating the properties which make transforms useful for noise reduction, beyond certain general properties (spatial localization, lack of aliasing, etc.) Nor is there adequate theory to describe the effects of estimating each subband coefficient individually in the presence of non-white noise.

In summary, the MCBSC algorithm presented here is certainly not the final word on multi-channel techniques which could be constructed based on the observations of the previous three chapters. However, it does demonstrate that these principles can guide a relatively simple attack on the problem which produces an effective algorithm. In fact, if one excludes a number of small-window techniques which cannot be effective on film grain (as shown in section 9.1) then MCBSC is the first technique which shows a significant improvement over single-channel approaches, without requiring unavailable *a-priori* information.

# Chapter 11

# Conclusions and Questions

## 11.1 Summary

This thesis began as an exploration of the unique properties of film grain (in still images) and a search for an algorithm which could effectively exploit these properties. Along the way, it became clear that the field of image noise reduction is somewhat confused and, while a few notable techniques have been reported, there is no general theory and no algorithm which is very effective in removing film grain. In particular the multi-channel literature is very sparse and not a single previously reported multi-channel technique is appropriate for film grain. All previous colour techniques either require unavailable information (like the Wiener and Kalman filters) or operate only over very small windows (like the various vector median filters).

A related problem was that, while the channels of a colour image are obviously highly "correlated", a precise definition and model of this phenomenon was lacking. In chapter 7 a series of simple experiments were conducted and the hypothesis formulated that the high-frequency components of each channel of a colour image are essentially identical. This observation formalizes the obvious notion that image "features" tend to be in the same place in all channels, and appears to be the first major advance in colour image models since it was observed long ago that almost all the information in a colour image is contained in the "luminance" component. Actually that observation and the proposed model are closely related, but formulation in terms of frequency content makes explicit the fact that colour images display simultaneous spectral

and spatial effects.

Then began, starting with chapter 8, a re-examination of the theory of noise reduction. It was shown that Bayesian estimation provides a unified framework in which to formulate noise reduction problems, in the sense that the required joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ embodies all possible prior knowledge of the image and the noise corrupting it. Further, it was argued that the minimum-variance Bayesian estimate is optimal in many ways, and provides a simple and useful formula for noise reduction.

Of course, the resulting estimation integral is only really solvable for very small numbers of variables, so a great deal of work was devoted in chapter 9 to finding ways around this problem. During this process, it was argued that dimensionality reduction is really the study of separable approximations to high-dimensional distributions. This viewpoint clarifies the relationship between correlation and statistical independence, and explains why, for example, transformations which aim to decorrelate the input variables are often effective. Finally, a series of experiments showed why independent channel processing cannot be effective in any colour space, and further arguments led the to model of equation 9.35, which implies that multi-channel noise reduction can be performed by operating on triplets of spatially transformed coefficients.

A model is only as good as its results, and chapter 10 used this model and the principles of the previous chapters to design a new noise reduction algorithm based on three-dimensional Bayesian estimation of subband coefficients. Along the way, a number of techniques and tricks were derived or developed which greatly increase the efficiency and robustness of the resulting algorithm. The proposed Multi-Channel Bayesian Subband Coring (MCBSC) technique is thus a practical technique, and experimental results confirm its efficacy as compared to single-channel approaches.

## 11.2 Future Work

As indicated at the end of the previous chapter, the MCBSC approach is certainly not the end-all of noise reduction approaches. Nor is the theoretical work presented in this thesis "complete" in any sense of the word. In many ways this thesis raises as many questions as it answers.

### 11.2.1 Dimensionality Reduction

The idea that dimensionality reduction is the process of constructing separable approximations is a useful idea but it is not prescriptive. That is, beyond certain fundamentals, almost nothing is known about the form of the ideal spatial transform for noise reduction. At least the following open questions remain:

- Are non-separable spatial transforms more effective for noise reduction, and if so why?
- Obviously any transform is limited by space/frequency resolution limits. What is the effect of each of these resolution parameters on noise reduction?
- Similarly, what is the ideal partitioning of the frequency plane for noise reduction?
- Although it has been shown that noise reduction may be performed by transforming each channel independently, it might be possible to obtain better performance by using a fully three-dimensional transform that operates both within and between channels. Such a transform would have to be non-separable, otherwise it would be equivalent to colour-space transformation, which has been demonstrated both theoretically and practically to be ineffective. Three dimensional non-separable transforms have not been well studied, so this area is wide open.
- Throughout this work, it has been claimed that aliasing of transform coefficients must be avoided, and this statement is backed up both theoretically and experimentally. However, this leads to overcomplete transformations, which are non-orthogonal. Such transformations do not correspond to a rotation of the coordinate axes and so it cannot really be claimed that a separable approximation to a high dimensional distribution is being constructed. Therefore, what is the effect of independent estimation of overcomplete transform coefficients on the Bayesian estimation equation? How does the level of redundancy affect the noise reduction process?

### 11.2.2 Distribution Modeling

In section 10.2.2 a relatively simple model of the 3D distribution of subband coefficients was presented. The results of chapter 7 indicate that such a model should always be sufficient in the sense that it will have the right general three-dimensional shape. While the proposed model was effective in practice, no alternatives were presented, and this leaves the following questions:

- How sensitive are noise reduction results on the distribution model and its accuracy? While initial experiments suggest that the answer is "not very", formal experiments need to be undertaken to determine the precise properties that are important for noise reduction.

- As presented, the MCBSC technique assumes that the coefficient distribution model is applicable across the entire image. It is well known that images display nonstationary statistical properties. How important is this effect, and how can the distribution model be varied over the image region to adapt to changes in image content?

- In the one dimensional case, simple functions such as clipped subtraction often provide very good approximations to the ideal Bayesian coring function. Are there equivalent simple functions for the multi-dimensional case? This is a topic of great practical importance, because the existence of a simple closed form approximation would remove the need for precomputation of lookup tables. Not only would this reduce the cost of the algorithm, but it would allow the adaptation to nonstationary statistics mentioned above. Simple approximations would also make hardware implementations practical.

### 11.2.3 Temporal Filtering

Throughout this work, the emphasis has been on still images. Temporal filtering of image sequences has been entirely ignored. This has resulted in an algorithm which can be applied to still images, but image sequences will certainly constitute a major use. In such sequences, the redundancy between frames is perhaps even greater than the redundancy between channels. Indeed, there exist several highly successful noise reduction approaches based on temporal

filtering, notably the hardware implementations produced by the British engineering firm of Snell & Wilcox.

Not only can temporal filtering further reduce the residual noise level, it is extremely important in the processing of sequences because it provides temporal stability. That is, while the current algorithm produces highly acceptable results when applied to still frames, the residual noise is completely uncorrelated between frames which makes it very visible when the entire sequence is displayed in real-time, as preliminary experiments have demonstrated. In aesthetic terms, it is not the absence of noise but the appearance of the absence of noise which is important, and lack of temporal stability makes residual noise very visible. Or, from a more practical point of view, video compression algorithms are quite sensitive to the differences between frames. Therefore some sort of temporal filtering seems to be required for image sequences.

Immediately it seems that temporal filtering approaches, which usually involve motion compensation followed by a noise reduction step akin to temporal coring, can be modified to use multi-channel estimation during the coring stage in an obvious way. However the present work suggests that perhaps a more subtle approach is possible. There has already been work in developing 3D transforms which are sensitive to motion between frames, such as the "garnet" transform [53]. Such a transform implicitly detects motion in the same way that the usual 2D subband transform implicitly detects edges. Motion estimation algorithms, because they must provide a single motion vector at each pixel, cannot hope to be robust and stable in the presence of noise, yet a multi-frame subband transform which is sensitive to motion will degrade gracefully under difficult situations. It is the overcompleteness of such transforms which is essential in this respect, because ambiguous motion will simply produce large responses in a number of differently oriented motion detection coefficients, in contrast to classical motion estimation algorithms which must choose a *single* displacement.

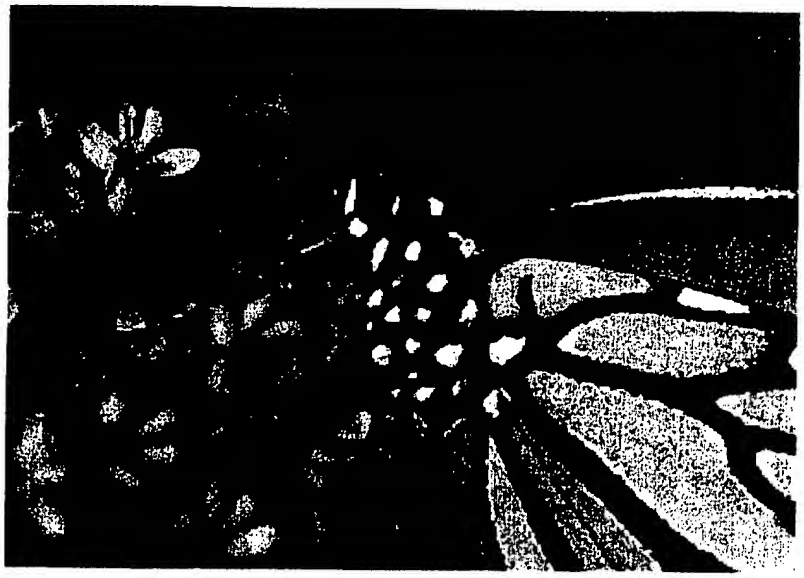
This suggests the following algorithm. First, all frames of the image sequence are decomposed, each channel independently, into an overcomplete coefficient set which describes both spatial and temporal edges of different orientations (note that the orientation of a temporal edge corresponds to direction and speed of motion.) Then, each coefficient of each band is estimated independently across the colour channels, using the 3D Bayesian estimation tech-

niques detailed in this thesis. Finally, inversion of the subband transform should give a set of spatially/spectrally/temporally filtered frames. Note that, to reduce memory requirements to practical levels, not all frames in a sequence need to be processed simultaneously, just as the finite spatial extent of subband filter kernels implies that, in principle, only a finite neighborhood of each pixel needs to be examined to produce a cleaned output. In much the same way, a temporal filtering algorithm based on subband decomposition need only examine enough adjacent frames to produce one cleaned frame at a time.

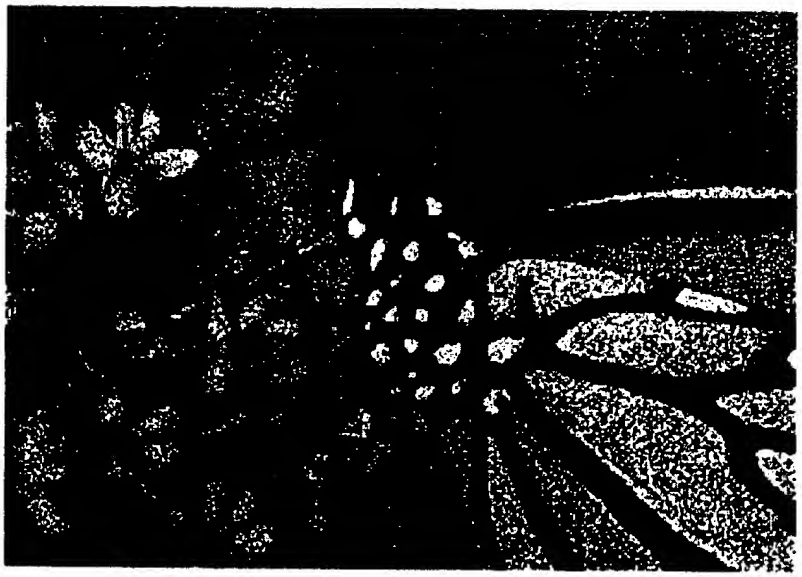
w

## 11.3

One can always ask for greater noise reduction, increased sharpness, better detail preservation, faster running time, and fewer artifacts, but the MCBSC algorithm is certainly an acceptable solution in that it meets all the above criteria quite well. In many ways, this algorithm is the best currently available technique for film grain reduction in particular and noise reduction in general for (still) colour images. Thus the original goal of this thesis – to find an effective grain reduction technique – has been met. However, as is often the case, the process of getting there has been at least as valuable as end result. Many interesting things about noise reduction have been learned in the search for an effective grain reduction technique, and hopefully the theory derived herein will prove to be a solid foundation for further research.

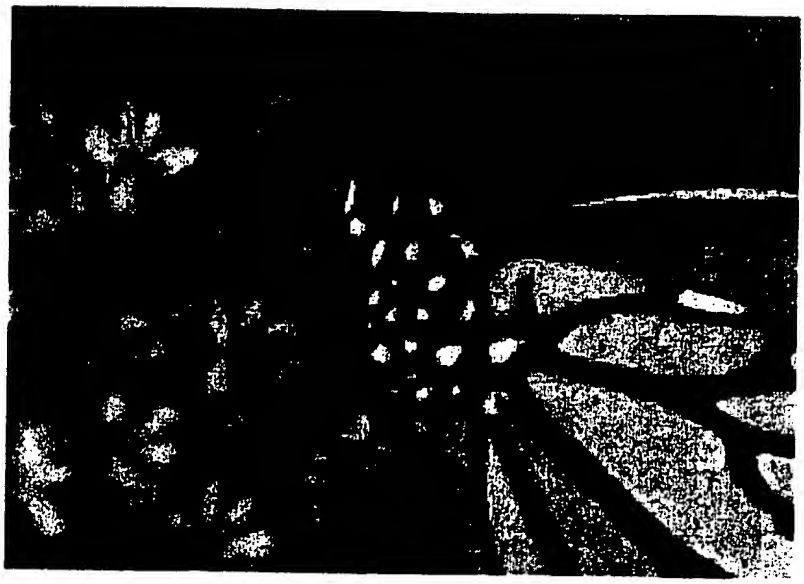


(a) original

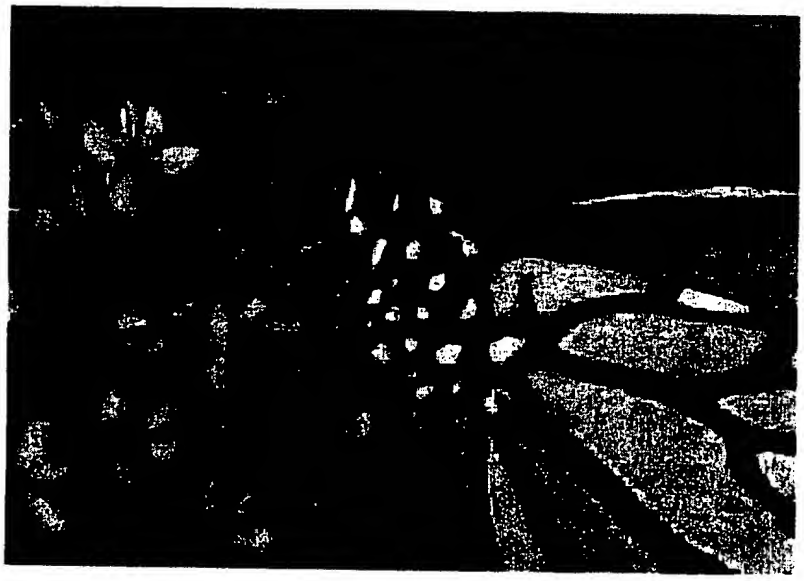


(b) added noise

Figure 10.6: Test case for single-channel vs. multi-channel noise reduction: (a) original image, (b) white noise added at a SNR of 8.5.



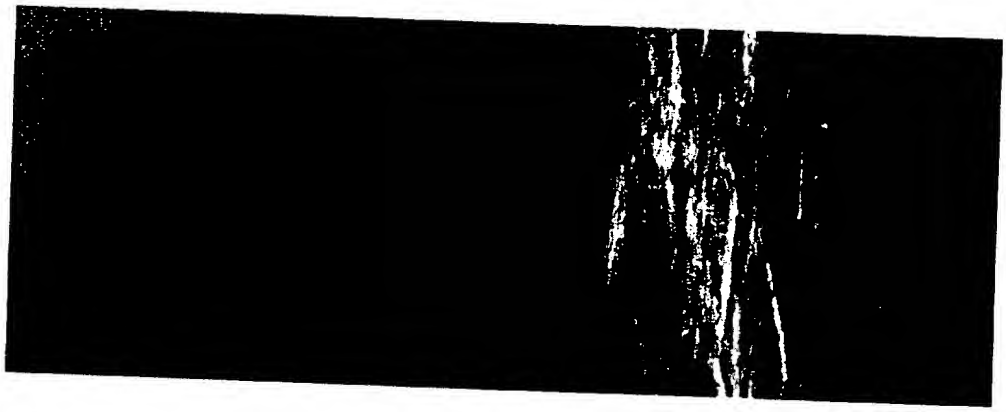
(a) single-channel



(b) multi-channel

Figure 10.7: Results of noise reduction by: (a) single-channel Bayesian subband coring and (b) multi-channel Bayesian subband coring. The multi-channel algorithm provides greater noise reduction and sharper results.

(a) original



(b) cleaned

Figure 10.8: Application of multi-channel Bayesian subband coring to real film grain.